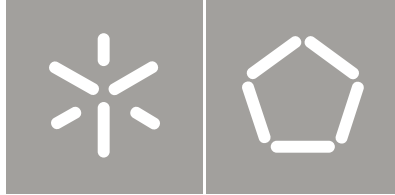


Universidade do Minho
Escola de Engenharia

Nelson Ricardo da Silva Marques

Eventos Sincronizados em
Sistemas Móveis Fisicamente Próximos



Universidade do Minho
Escola de Engenharia

Nelson Ricardo da Silva Marques

Eventos Sincronizados em
Sistemas Móveis Fisicamente Próximos

Tese de Mestrado
Ciclo de Estudos Integrados Conducentes ao
Grau de Mestre em Engenharia de Comunicações

Trabalho efetuado sob a orientação do
Professor Doutor Filipe Meneses

Agradecimentos

A conclusão deste estudo não seria possível sem a ajuda de algumas pessoas, a quem quero apresentar aqui alguns agradecimentos.

Quero agradecer ao Professor Doutor Filipe Meneses pela disponibilidade, empenho e dedicação que sempre demonstrou, bem como pelas sugestões e correções. Foi um privilégio realizar esta investigação sob a sua orientação.

Aos meus Pais e à minha Irmã, agradeço todo o esforço empregue durante estes anos. Não existem palavras para expressar o quanto são importantes na minha vida.

À Beatriz e ao Silvino, pela constante demonstração de carinho e afeto.

À Andreia, agradeço por tudo. Mas em especial pelo apoio incondicional, admiração, paciência e perseverança.

Finalmente, mas não menos importante, quero agradecer a todos os meus Amigos.

A todos, manifesto aqui a minha sincera gratidão,

Obrigado!

Resumo

O desenvolvimento de aplicações para dispositivos móveis tornou-se uma área de grande interesse, em resultado da enorme disponibilidade de dispositivos no mercado e da sua crescente integração com tecnologias de comunicação sem fios.

O objetivo deste trabalho consiste na criação de um sistema destinado a proporcionar a realização de um determinado efeito coordenado, entre os telefones móveis dos espectadores presentes num evento.

A investigação seguiu o método usual utilizado em estudos desta área de investigação. Foi realizada uma revisão do estado de desenvolvimento de aplicações para dispositivos móveis e das tecnologias que integram. De seguida, foram combinadas e discutidas diferentes tecnologias para resolver os desafios do trabalho apresentado. Os resultados sugerem a existência de algumas lacunas no Bluetooth que, apesar de ser uma tecnologia bastante desenvolvida, requer demasiado tempo para o estabelecimento de ligações com outros dispositivos.

Foi construído um sistema, designado de SyncEvent, capaz de realizar um efeito coordenado, através de um grupo de utilizadores que se encontrem fisicamente próximos. Para um utilizador participar num evento é necessário que o sistema possua a informação da sua localização. Sendo assim, considera-se uma abordagem onde os dispositivos podem ser localizados de forma explícita ou estimada.

O trabalho realizado ao longo desta dissertação foi desenvolvido em ambiente laboratorial, realizando alguns testes utilizando dois dispositivos móveis com diferentes características técnicas.

Ao longo da investigação realizada nesta dissertação, foi apresentado numa conferência internacional, o trabalho que se encontra em apêndice B.

Abstract

The development of applications for mobile devices has become an area of great interest, as a result of the enormous availability of devices in the market and its increasing integration with wireless communication technologies.

The purpose of this study is to create a system to provide the performance of a particular effect coordinated between the mobile phones of spectators present at an event.

The research has followed the usual method used in studies of this area of research. A review of the status of development of the applications for mobile devices and technologies that integrate. Then it was combined and discussed various technologies to solve the challenges of the presented work. The results suggested the existence of some gaps in Bluetooth which despite being a well developed technology, requires too much time to establish connection to other devices.

It was built a system, designated as SyncEvent, able to carry out a coordinated effect, through a group of users who are physically near. It is necessary that the system have information of the user's location, so that, they can participate in the event. Therefore, we propose an approach where the devices can be found in a explicitly or estimated way.

The work done along this thesis has been developed in the laboratory by performing some tests using two mobile devices with different technical characteristics.

Along the research conducted in this dissertation, the work that is in the Appendix B, was presented at an internacional conference.

Índice de conteúdo

Agradecimentos	iii
Resumo	v
Abstract	vii
Índice de conteúdo	ix
Índice de figuras	xiii
Índice de tabelas	xv
Acrónimos	xvii
1 Introdução	1
1.1 Enquadramento	1
1.2 Objetivos	2
1.3 Estrutura da dissertação	3
2 Estado da arte	5
2.1 Dispositivos móveis	5
2.1.1 Telefones móveis	6
2.2 Desenvolvimento de aplicações móveis	12
2.2.1 Mercado de dispositivos móveis	12
2.2.2 Modelos de negócio	13
2.2.3 Sistemas operativos móveis	13
2.2.4 Outras plataformas de desenvolvimento	18
2.3 Soluções propostas anteriormente	23

2.4	Conclusões	23
3	Tecnologias de comunicações sem fios	25
3.1	IrDA - Infrared Data Association	25
3.2	RFID - Radio Frequency Identification	26
3.3	NFC - Near-Field Communication	27
3.4	ZigBee	28
3.5	Rede móvel celular	28
3.6	IEEE 802.11	29
3.6.1	Arquitetura do sistema	30
3.6.2	Camada física	33
3.6.3	Camada MAC	33
3.7	Bluetooth	37
3.7.1	Arquitetura	37
3.7.2	Camada física	38
3.7.3	Funcionamento básico	38
3.7.4	Formato dos pacotes	40
3.7.5	Estabelecimento da ligação	40
3.7.6	Bluetooth v4.0	41
3.8	Conclusões	41
4	Estudo do Problema	43
4.1	Abordagem sem infraestrutura	43
4.1.1	Estabelecimento da ligação	45
4.1.2	Poluição em redes Bluetooth	45
4.1.3	Criação de <i>scatternet</i>	46
4.1.4	Partição da rede	47
4.1.5	Diversidade de dispositivos	47
4.1.6	Sincronização	48
4.2	Abordagem com infraestrutura	49
4.2.1	Operação básica	50
4.2.2	Complexidade e custo	50
4.2.3	Localização dos dispositivos	51
4.2.4	Difusão dos eventos	51
4.3	Abordagem alternativa	52

4.3.1	Vantagens	52
4.3.2	Limitações	53
4.4	Conclusões	54
5	Especificação do Sistema	55
5.1	Arquitetura global	55
5.2	Arquitetura do projeto	57
5.3	Camadas do sistema	58
5.3.1	Módulo de mapeamento	59
5.3.2	Módulo de localização	64
5.3.3	Módulo de sincronização	65
5.4	Protocolo de comunicação	66
5.4.1	Localização explícita de dispositivos	67
5.4.2	Localização implícita de dispositivos	68
5.4.3	Participação num evento	69
5.4.4	Formato das mensagens	70
5.5	Conclusões	72
6	Implementação	73
6.1	Descrição do cenário	73
6.2	Escolha do sistema operativo	74
6.3	Escolha da linguagem de programação	79
6.4	Protocolo de comunicação	79
6.5	Elementos do sistema	80
6.5.1	Elementos do servidor	80
6.5.2	Elementos do dispositivo móvel	82
6.6	Dispositivos móveis utilizados	83
6.6.1	Samsung Galaxy S	84
6.6.2	Samsung Galaxy Mini	84
6.7	Resultados obtidos	85
7	Conclusões e trabalho futuro	89
7.1	Conclusões do projeto	89
7.2	Limitações da aplicação desenvolvida	90
7.3	Evolução no mecanismo de posicionamento	91

7.4 Trabalho futuro	91
Referências e bibliografia	93
A Protocolo de Comunicação	101
B Contributo	103

Índice de figuras

2.1	Interfaces de comunicação dos <i>smartphones</i>	8
3.1	Camadas da especificação IEEE 802.11	31
3.2	Arquitetura de uma rede IEEE 802.11 baseada em infraestrutura . . .	32
3.3	Arquitetura de uma rede <i>ad-hoc</i> IEEE 802.11	32
3.4	Camada de controlo de acesso ao meio do IEEE 802.11	34
3.5	Mecanismo básico de controlo de acesso ao meio do IEEE 802.11 . . .	35
3.6	Mecanismo RTS/CTS do IEEE 802.11	36
3.7	Mecanismo de acesso ao meio livre de contenção (PCF)	36
3.8	Arquiteturas possíveis do Bluetooth	37
3.9	Topologias de comunicação Bluetooth	39
3.10	Formato genérico dos pacotes Bluetooth	40
3.11	Estabelecimento de uma ligação Bluetooth	41
4.1	Abordagem sem infraestrutura	44
4.2	Abordagem com infraestrutura	49
4.3	Grafo de proximidade entre dispositivos móveis.	53
5.1	Arquitetura global	56
5.2	Arquitetura do projeto	57
5.3	Camadas do sistema	58
5.4	Exemplo de um estádio	60
5.5	Matriz de mapeamento do estádio	61
5.6	Localização de um dispositivo de forma estimada	65
5.7	Localização explícita de dispositivos	68
5.8	Localização implícita de dispositivos	69
5.9	Participação num evento	70

5.10	Formato geral das mensagens do SyncEvent	70
5.11	Formato da mensagem de localização do SyncEvent	71
5.12	Formato da mensagem de participação em eventos do SyncEvent . . .	72
6.1	Exemplo de uma onda Mexicana ou <i>La Ola</i>	74
6.2	Arquitetura global do Android	76
6.3	Simplificação das relações entre as classes do servidor	81
6.4	Simplificação das relações entre as classes da aplicação móvel	82
6.5	Samsung i9000 Galaxy S 8GB	84
6.6	Samsung S5570 Galaxy Mini	85
6.7	Aplicação móvel do SyncEvent	86
6.8	Servidor do SyncEvent	87

Índice de tabelas

2.1	Vendas Mundiais de <i>smartphones</i> em 2010	13
3.1	Classes de dispositivos Bluetooth	38
5.1	Tipos de mensagens definidos no SyncEvent	71
6.1	Versões do sistema operativo Android	75
6.2	Especificações técnicas do Samsung i9000 Galaxy S 8GB	84
6.3	Especificações técnicas do Samsung S5570 Galaxy Mini	85

Acrónimos

API	Application Programming Interface
BR	Basic Rate
BREW	Binary Runtime Environment for Wireless
BSA	Basic Service Area
BSS	Basic Service Set
CDMA	Code Division Multiple Access
CLDC	Connected Limited Device Configuration
CSMA/CA	Carrier Sense Multiple Access/Collision Avoidance
DCF	Distributed Coordination Function
DLNA	Digital Living Network Alliance
DS	Distribution System
DSSS	Direct Sequence Spread Spectrum
EDGE	Enhanced Data rates for GSM Evolution
EDR	Enhanced Data Rate
ETSI	European Telecommunications Standards Institute
FHSS	Frequency Hopping Spread Spectrum
GFSK	Gaussian Frequency Shift Keying
GPRS	General Packet Radio Service
GPS	Global Positioning System
GSM	Global System for Mobile communications
HCF	Hybrid Coordination Function
HCI	Host Controller Interface
HSCSD	High Speed Circuit-Switched Data
HSPA	High Speed Packet Access

IBSS	Independent Basic Service Set
IR	Infra Red
IrDA	Infrared Data Association
ISM	Industrial-Scientific-Medical
J2ME	Java 2 Micro Edition
JCP	Java Community Process
JSR	Java Specification Request
JVM	Java Virtual Machine
LTE	Long Term Evolution
MAC	Medium Access Control
MANET	Mobile Adhoc Networks
MIDP	Mobile Information Device Profile
MMS	Multimedia Messaging Service
MTP	Media Transfer Protocol
NFC	Near Field Communication
OFDM	Orthogonal Frequency-Division Multiplexing
PAN	Personal Area Network
PC	Personal Computer
PCF	Point Coordination Function
PDA	Personal Digital Assistant
PIN	Personal Identification Number
QoS	Quality of Service
RFID	Radio Frequency IDentification
SMS	Short Message Service
TDD	Time Division Duplexing
UMTS	Universal Mobile Telecommunications System
WAP	Wireless Application Protocol
WLAN	Wireless Local Area Network
WML	Wireless Markup Language
WWW	World Wide Web

Capítulo 1

Introdução

Este capítulo contém uma introdução à criação de eventos sincronizados em sistemas móveis fisicamente próximos. Num primeiro instante apresenta-se uma visão global do trabalho através do respetivo enquadramento, seguindo-se os objetivos e principais desafios inerentes ao trabalho proposto. Na parte final do capítulo encontra-se descrita a organização da presente dissertação.

1.1 Enquadramento

Encontrar alguém que num dado momento não tenha consigo um telefone móvel é, cada vez mais, uma exceção. Com a evolução tecnológica, num curto espaço de tempo, os telefones móveis passaram de dispositivos destinados a assegurar exclusivamente a comunicação de voz, para sistemas abertos, capazes de assegurar as comunicações de voz e vídeo, o envio/receção de mensagens, a execução de aplicações, acesso à Internet, etc.

Hoje em dia, os *smartphones*, ou “telefones inteligentes” como muitas vezes são designados, representam o grupo preponderante de telefones móveis e apresentam já um extenso conjunto de serviços e aplicações. Uma das grandes vantagens dos *smartphones* é a mobilidade que oferecem, podendo o acesso à informação ser realizado em qualquer local e a qualquer momento. Por outro lado, podem ser aproveitadas as oportunidades que o *cloud computing*¹ possa oferecer (ex. aplicações escaláveis).

¹ Em *cloud computing*, recursos como processador, memória e armazenamento, não se encontram fisicamente presentes nos dispositivos dos utilizadores. Em vez disso, as aplicações e os dados encontram-se acessíveis, em qualquer altura e com qualquer dispositivo, através da Internet. [Kumar & Lu, 2011]

O *smartphone*, mais do que qualquer outra tecnologia que surgiu na última década, é o que mais tem alterado o comportamento das pessoas [Romero, 2011]. Com os telefones móveis tradicionais os utilizadores passavam a maior parte do tempo em chamadas de voz ou no envio de mensagens. Num *smartphone*, os serviços de comunicação básicos dão lugar à navegação na Internet, ao *e-mail* e ao entretenimento. Quando se fala em *smartphones* é inevitável lembrar da quantidade crescente de pessoas que os utilizam para o acesso a redes sociais como LinkedIn, Twitter, Facebook e Google+.

O computador já não é a plataforma predileta, e surgem cada vez mais *gadgets*² como *smartphones* e *tablets* que disponibilizam acesso à Internet e instalação de aplicações. O desenvolvimento de aplicações para dispositivos móveis tornou-se numa área de grande interesse para a computação devido não só à enorme disponibilidade de dispositivos no mercado, mas também em virtude da sua crescente integração com tecnologias de comunicação sem fios. Esses dispositivos fazem cada vez mais parte do quotidiano das pessoas e são exemplo da universalização da tecnologia e da informação.

Quando um recinto se enche de espectadores para assistirem a um evento é habitual que as pessoas efetuem algumas coreografias e entoem cânticos, mostrando o seu apoio e agrado aos desportistas ou artistas. Por exemplo, num estádio de futebol é usual as pessoas levantarem os braços, de forma coordenada, produzindo um efeito visual chamado de “onda”.

1.2 Objetivos

O objetivo do trabalho é desenvolver um sistema que permita produzir um determinado efeito coordenado entre os telefones móveis dos espectadores presentes num evento. Por exemplo, reduzindo a iluminação de um estádio os espectadores podem levantar no ar os seus telefones móveis e ver surgir uma “onda”. A diferença para o modelo de braços no ar é que agora a onda é feita, por exemplo, acendendo de forma coordenada, a iluminação do visor/teclado dos dispositivos móveis.

A realização do trabalho proposto apresenta vários desafios:

- Difundir em tempo útil um evento por um elevado número de telefones móveis;

² *Gadgets* são dispositivos ou aplicações informáticas que se revelam úteis para determinadas tarefas.

- Conseguir que o evento ocorra de forma sincronizada;
- Obter a localização dos utilizadores;
- Lidar com a heterogeneidade dos dispositivos em termos de sistemas operativos, capacidades de processamento e de comunicação.

1.3 Estrutura da dissertação

Este documento encontra-se estruturado em vários capítulos. Sendo que neste capítulo primogénito é apresentada uma visão generalista do contexto em que o presente trabalho se insere e principais desafios intrínsecos.

O capítulo 2 contém uma primeira fase de investigação e de ideias, onde se encontra um resumo do aparecimento e consequente evolução histórica dos principais dispositivos móveis, alguns sistemas operativos, suas vantagens e desvantagens, bem como uma breve descrição do seu funcionamento. São também focadas algumas soluções comerciais existentes para o desenvolvimento de aplicações móveis.

No capítulo 3 são caracterizadas e comparadas algumas das interfaces de comunicação presentes nos dispositivos móveis.

No capítulo 4 encontra-se uma conceptualização do problema. Nesta parte da dissertação apresentam-se algumas abordagens para a resolução do problema e as razões para a seleção de uma em particular.

Na fase de especificação (capítulo 5), foi concebida uma arquitetura que consentisse os requisitos necessários. Aqui é apresentada a arquitetura do projeto e descrito o protocolo de comunicação estabelecido entre os intervenientes do sistema.

O capítulo 6, de natureza mais prática, apresenta o protótipo da aplicação para a criação de eventos sincronizados em sistemas móveis fisicamente próximos. Neste capítulo é descrito o cenário de teste, são fundamentadas as opções realizadas e apresentados os resultados obtidos. Esses resultados além de comprovarem toda a pesquisa realizada na primeira fase, são ainda utilizados para rever possíveis melhorias no sistema implementado.

Por fim, no sétimo e último capítulo, são expostas as conclusões obtidas de modo transversal com o trabalho desenvolvido e, acrescentadas ainda, algumas orientações para uma investigação futura.

Capítulo 2

Estado da arte

A área da computação móvel passou por uma sucessão gradual de transformações nos últimos anos. Em particular, o mercado de dispositivos, é dominado por um conjunto de diferentes plataformas tecnológicas, incluindo diferentes sistemas operativos e diferentes plataformas de desenvolvimento de software, resultando numa variedade de soluções no mercado. Este capítulo apresenta uma perspetiva geral do estado da arte, permitindo obter o conhecimento da situação atual do mercado de dispositivos móveis e do desenvolvimento de aplicações direcionadas a esses.

2.1 Dispositivos móveis

Nos últimos anos, apesar da grande utilização de dispositivos móveis como *notebooks*, *netbooks*, Personal Digital Assistants (PDAs), câmaras digitais, leitores mp3, entre outros, o telemóvel é o dispositivo mais comum, sendo considerado indispensável para a maioria das pessoas. Devido à enorme capacidade de comunicação com familiares, amigos, etc., não só através de chamadas e Short Message Service (SMS) mas até pelo envio de *e-mails*. Hoje em dia os telemóveis integram câmara fotográfica e de vídeo, leitor mp3 e rádio, tudo num dispositivo móvel cada vez mais pequeno. Além de ajudarem a manter o contacto com outras pessoas, ajudam a gerir o quotidiano dos utilizadores com funcionalidades como lista de contactos, calendário, alarmes, aplicações empresariais, etc. Consequentemente, os telemóveis são transportados diariamente com os utilizadores, encontrando-se perto e acessíveis em qualquer momento, independentemente da atividade do utilizador.

Classificar e caracterizar os equipamentos móveis disponíveis no mercado é uma

tarefa complexa. O aparecimento de dispositivos móveis, que integram várias funcionalidades (câmara fotográfica e de vídeo, leitor mp3, rádio, aplicações, *office*, agenda, etc.), veio complicar a disposição dos dispositivos numa categoria específica.

Segundo Livingston [2004], os dispositivos móveis podem ser divididos em duas categorias. A primeira categoria engloba os telefones móveis e os PDAs, enquanto que a segunda categoria contém o grupo dos *two-way pagers*¹. A presunção era englobar no primeiro grupo *tablets*, *notebooks* e *laptops*. No entanto, apesar de permitirem uma grande mobilidade, não acompanham os utilizadores no seu quotidiano da mesma forma que os telefones móveis, PDAs e *pagers*.

A seguir identificam-se as características do principal grupo de dispositivos móveis e as várias formas de utilização.

2.1.1 Telefones móveis

Os telefones móveis surgem hoje em dia em vários modelos da mesma marca e com uma abundância de recursos. Livingston [2004] sugere dividir a generalidade de telefones móveis em três categorias: telefones móveis com capacidade de navegação na web, telefones móveis com capacidade de extensão e *smartphones*. As capacidades que definem essas categorias são acumulativas, ou seja, os telefones móveis com capacidade de extensão possuem todas as capacidades dos telefones móveis com capacidade de navegação na web, e por sua vez os *smartphones* possuem todas as capacidades dos telefones móveis extensíveis.

Telefones móveis com capacidade para a web

Os telefones móveis com capacidade de navegação na web, ou na língua Inglesa *web-enabled phones*, possuem quatro serviços principais: voz, mensagens escritas, navegação na web e um conjunto de diversas aplicações como calendário, calculadora, jogos, entre outros. O serviço de voz requer pouca ou nenhuma explicação. A grande maioria das pessoas, utiliza os seus telefones móveis exclusivamente para comunicação de voz [Livingston, 2004]. Algumas das características relacionadas com o serviço de voz são o armazenamento do número de outros utilizadores, o serviço de *voicemail*, entre outros.

¹ *Two-way pagers* são dispositivos móveis com capacidade para enviar/receber *e-mails* e mensagens de texto.

Alguns telefones (especialmente aqueles que possuem câmara integrada), além de disponibilizam o envio de pequenas mensagens de texto, permitem ainda o envio de mensagens multimédia. Também na categoria de mensagens, a maioria dos dispositivos com capacidade de navegação na web possui ainda capacidades de *e-mail*.

Além do serviço de voz e de mensagens, este grupo de dispositivos móveis possui normalmente um *browser* simples, que permite ao utilizador visualizar alguns conteúdos da web. Estes *browsers* são baseados em Wireless Application Protocol (WAP). Isto significa que o acesso à rede é realizado através de um servidor WAP, e que apenas podem aceder a conteúdo Wireless Markup Language (WML), ou seja, conteúdo especificamente codificado para *browsers* WAP.

Telefones móveis com capacidade de extensão

Extensible Phones, ou telefones móveis com capacidade de extensão, possuem todas as capacidades dos telefones móveis com capacidade de navegação na web e, em adição, permitem aos seus utilizadores alargar as suas funcionalidades através do *download* e da instalação de software. O conjunto básico de serviços que este grupo de dispositivos disponibiliza é o seguinte:

- Voz
- Mensagens escritas
- Navegação na web
- Extensibilidade
- Outros diversos

Existem dois grupos principais de *extensible phones*. O primeiro e mais comum são os telefones baseados na plataforma Java 2 Micro Edition (J2ME). O software deste grupo de dispositivos inclui um módulo de gestão de aplicações e uma máquina virtual, o que permite a instalação e utilização de programas baseados em J2ME.

O segundo grupo de *extensible phones* são baseados na plataforma Binary Runtime Environment for Wireless (BREW). Esta plataforma foi desenvolvida pela empresa Qualcomm para criação e distribuição de aplicações para telefones móveis. A distribuição das aplicações é realizada através de uma aplicação da mesma empresa chamada de BREW Shop. Esta aplicação encontra-se instalada de origem em todos

os dispositivos que suportam esta tecnologia, permitindo a compra e *download* de aplicações aprovadas pela operadora.

Smartphones

Os *smartphones* são a principal tendência no mercado de telecomunicações. Segundo a IDC, empresa que analisa o mercado tecnológico, nomeadamente os telefones móveis, os *smartphones* já vendem mais que computadores pessoais². Este acontecimento histórico, aconteceu pela primeira vez no último trimestre de 2010, onde foram vendidos mais de 100 milhões de *smartphones* contra os 92,1 milhões de Personal Computers (PCs) [Ziegler, 2011].

Os *smartphones* integram os serviços de telecomunicações e de Internet num único dispositivo, que combina a portabilidade dos telefones móveis com o poder de computação e de comunicação dos PCs [Guo et al., 2007]. Como ilustrado na figura 2.1, os *smartphones* encontram-se integrados tanto na rede móvel como na Internet.

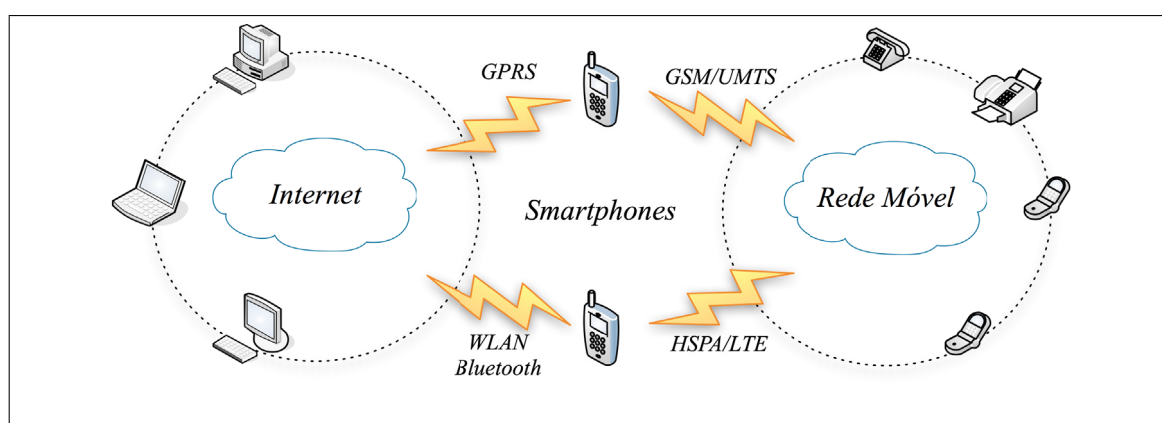


Figura 2.1: Interfaces de comunicação dos *smartphones* [Guo et al., 2007].

Outras razões para esta tendência é a facilidade e o baixo custo de integração dos serviços de Internet e de telecomunicações. Uma vez que os *smartphones* são, hoje em dia, tão poderosos como alguns computadores pessoais de alguns anos atrás, os seus sistemas operativos evoluíram bastante.

A evolução verificada nos telefones móveis é, de certa forma, similar à que se verifica em outras áreas (ex.: computadores). O constante progresso resulta essencialmente da evolução na eletrónica que permitiu obter melhores *displays*, mais

² A expressão “computador pessoal” ou PC (abreviatura da expressão de origem Inglesa “personal computer”) é utilizada para designar computadores de mesa ou portáteis com diferentes características tecnológicas.

capacidade de armazenamento a menor custo, processadores com maior capacidade e baterias melhores (sendo este o ponto onde se verifica uma menor evolução). A melhoria no hardware foi acompanhada por uma evolução constante no software.

Hoje em dia, o mercado é dominado por um grande número de diferentes plataformas tecnológicas, incluindo diferentes sistemas operativos e diferentes plataformas de desenvolvimento de software, resultando numa variedade de soluções concorrentes no mercado. Mais à frente faz-se uma análise comparativa dessas plataformas tecnológicas e identificam-se as vantagens e desvantagens das principais plataformas que dominam o mercado nos dias de hoje.

Embora os detalhes de *design* e de funcionalidades variem entre esses sistemas operativos, todos eles partilham as seguintes características [Guo et al., 2007]:

- Acesso à rede móvel através de Global System for Mobile communications (GSM), Universal Mobile Telecommunications System (UMTS) e/ou Code Division Multiple Access (CDMA).
- Acesso à Internet através de diversas interfaces de rede, tais como Bluetooth, General Packet Radio Service (GPRS) e Wireless Local Area Network (WLAN);
- *Multitasking* para execução de várias aplicações em simultâneo;
- Sincronização de dados com computadores pessoais;
- Application Programming Interfaces (APIs) de código aberto para o desenvolvimento de aplicações.

A distinção entre as várias categorias apresentadas por Livingston [2004] é cada vez mais difícil. Atualmente, os *smartphones* representam o grupo preponderante de “telefones móveis”. A distância entre os diversos modelos é, hoje em dia, realizada através dos diferentes sistemas operativos, das diferentes capacidades de armazenamento e processamento, das diferentes resoluções dos *displays*, etc. No entanto, as principais características de um *smartphone* encontram-se em praticamente todos os modelos existentes no mercado.

Potencial dos telefones móveis

O grande poder computacional dos processadores, as melhorias nos sistemas operativos e a popularidade da banda larga móvel, tornaram os telefones móveis em sofisticados dispositivos de computação móvel [Wong, 2010].

De seguida, identificam-se as principais características dos telefones móveis modernos que beneficiam a computação móvel.

Banda larga móvel: As redes móveis evoluíram de sinais analógicos para digitais e de redes de comutação de circuitos para redes de comutação de pacotes. A velocidade de transferência de dados aumentou consideravelmente de 9.6 kbps no serviço básico GSM para 171.2 kbps no GPRS, de 384 kbps com o Enhanced Data rates for GSM Evolution (EDGE) para 2 Mbps com o UMTS e, atualmente, o Long Term Evolution (LTE) já oferece velocidades de *download* que podem chegar até 300 Mbps [Ghosh et al., 2010]. Juntamente com a evolução das infraestruturas de comunicação, o número de utilizadores de banda larga móvel aumentou substancialmente nos últimos anos, não só graças à facilidade de utilização, mas também pelos baixos preços praticados, e ainda pela conveniência do acesso à Internet em qualquer local. A disponibilidade de acesso à Internet tornou-se na principal preocupação dos fabricantes de dispositivos móveis facilitando a evolução da computação móvel.

Poder de processamento: Os processadores dos *smartphones* evoluíram de tal forma que hoje em dia podem ser utilizados em *laptops* ou *netbooks*. Por exemplo, o processador Snapdragon de 1 GHz da empresa de telecomunicações Qualcomm, dos Estados Unidos da América, pode ser utilizado em dispositivos com ecrãs até 12 polegadas com uma resolução até 1440x900 pixels [Wong, 2010]. Vários *smartphones* recentes utilizam este processador, incluindo o HTC HD2 e o Sony Ericsson's Xperia X10. Os processadores dos dispositivos móveis também já entraram na era dos processadores com vários núcleos. No final de 2010 foi lançado, pela multinacional Sul-Coreana LG, o primeiro *smartphone* (LG Optimus 2X) com um processador *dual-core* [Ziegler, 2010]. Durante a realização desta dissertação foi anunciado o iPhone 4S que contém o chip dual-core A5 da Apple. Segundo o site oficial, o iPhone 4S possui o dobro do poder de processamento do que o seu antecedente. Esta evolução dos processadores de dispositivos móveis veio facilitar ainda mais a sofisticação das aplicações móveis.

Grande capacidade de armazenamento: Há alguns anos, os dispositivos móveis possuíam uma capacidade de armazenamento interno mínima (inferior a 4Mb). Atualmente, oferecem capacidades de armazenamento até 64GB como é o caso do recente Samsung Galaxy S II [Samsung, 2011]. Considerando a evolução contínua nas tecnologias de armazenamento, tais como pequenas unidades de disco e memória *flash*, provavelmente no futuro os *smartphones* irão oferecer ainda mais capacidade

de armazenamento de dados. Tendo em conta que o acesso à Internet é cada vez mais usual em dispositivos móveis, é também possível aceder a dados guardados em outros pontos da rede (como por exemplo *cloud computing* e servidores empresariais).

Amplas bibliotecas para o desenvolvimento de aplicações: Plataformas móveis modernas, tais como iOS e Android, incluem tipicamente uma ampla gama de bibliotecas e APIs para os programadores. Além das bibliotecas para o acesso a sensores internos como acelerómetros, Global Positioning System (GPS) ou magnetómetros, existem bibliotecas de comunicação direcionadas para Internet, para reprodução de multimédia e para controlo de mapas. Essas bibliotecas e APIs permitem que os programadores desenvolvam novas aplicações e controlem o telefone com grande liberdade e flexibilidade.

Interfaces de utilização avançadas: Nos últimos anos, deu-se uma grande evolução no desenvolvimento de interfaces para os telefones móveis. A evolução mais perceptível é a tecnologia *multitouch* [Wong, 2010]. Esta tecnologia consiste num ecrã sensível ao toque que reconhece vários toques em simultâneo, possibilitando uma interação mais intuitiva através das mãos dos utilizadores. Por exemplo, para aumentar o *zoom* numa foto utilizando um interface *multitouch*, o utilizador pode tocar na foto com dois dedos e de seguida afastá-los. Muitos telefones móveis (e muitos outros dispositivos) utilizam esta tecnologia, e os programadores de software também a utilizam para várias aplicações de *design*.

Grande penetração: O número de utilizadores da rede móvel em todo o mundo chegou a 4 mil milhões no final de 2009, e é esperado que cresça mil milhões por ano [Wong, 2010]. Algumas pessoas possuem vários telefones móveis, um exclusivamente para utilização pessoal, e outro para utilização profissional, ou de forma a possuírem várias redes para reduzir os gastos de comunicação. Além disso, a maioria dos proprietários de telefones móveis nunca saem de casa sem o seu telefone móvel. Estes dispositivos disponibilizam a forma mais conveniente para as pessoas lidarem com as suas tarefas diárias, como gestão de informação pessoal ou mesmo na realização dos seus negócios. Os telefones móveis encontram-se cada vez mais infiltrados, podendo realizar tarefas de computação distribuída e executar aplicações baseadas na sua localização.

2.2 Desenvolvimento de aplicações móveis

Nos últimos anos, tem-se assistido a uma mudança radical pela introdução e pelo desenvolvimento dos *smartphones*. No entanto, o mercado de dispositivos é dominado por um conjunto de diferentes plataformas tecnológicas, incluindo diferentes sistemas operativos e diferentes plataformas de desenvolvimento de software, resultando numa variedade de soluções no mercado.

Hoje em dia, o mercado disponibiliza aos programadores várias tecnologias para desenvolvimento de aplicações móveis. Devido às várias combinações possíveis de linguagens de programação, sistemas operativos e dispositivos móveis, decidir que linguagem de programação utilizar é uma tarefa complicada. Atualmente encontram-se disponíveis inúmeras plataformas de desenvolvimento para dispositivos móveis, incluindo ambientes nativos como Symbian, Android, Blackberry, iOS, Windows Phone e webOS, e plataformas de desenvolvimento como Python, Lazarus, Brew, Java Mobile Edition, .NET Compact Framework e Flash Lite.

Normalmente, os principais fatores que definem a escolha da plataforma de desenvolvimento é a formação técnica pessoal e as tendências do mercado [Hammershoj et al., 2010]. Por outro lado, a escolha deve ser realizada considerando as necessidades do programador (que funcionalidades necessita de implementar) e as facilidades que cada linguagem apresenta.

De forma a verificar o estado atual e as tendências em plataformas de desenvolvimento, apresenta-se de seguida uma análise de mercado e comparam-se as plataformas mais populares nos dias de hoje.

2.2.1 Mercado de dispositivos móveis

Segundo um estudo realizado pela Gartner [Gartner, 2011], em fevereiro de 2011, o número de vendas de dispositivos móveis inteligentes, vulgarmente designados por *smartphones*, totalizou 1.6 mil milhões de unidades em 2010, tendo um crescimento de 31.8 pontos percentuais em relação ao ano anterior [Rao, 2011].

Como é possível visualizar na tabela 2.1, o Symbian, sistema operativo utilizado nos telefones móveis da Nokia, continua a liderar o *ranking* dos sistemas operativos para dispositivos móveis. Em termos de vendas por sistema operativo, o Android cresceu 888.8% em 2010 e alcançou a segunda posição. Apesar de muitos oferecerem o título do sistema operativo mais vendido ao Android, segundo Gartner [2011], o mer-

Tabela 2.1: Vendas Mundiais de *smartphones* em 2010 [Rao, 2011].

Empresa ou Sistema Operativo	Unidades Vendidas em 2010 (M)	Quota de Mercado em 2010 (%)	Unidades Vendidas em 2009 (M)	Quota de Mercado em 2009 (%)
Symbian	11,576.7	37.6	80,878.3	46.9
Android	67,224.5	22.7	6,798.4	3.9
RIM	47,451.6	16.0	34,345.6	19.9
iOS	46,598.3	15.7	24,889.7	14.4
Microsoft	12,378.2	4.2	15,031.0	8.7
Outros	11417.4	3.8	10,432.1	6.1
Total	296,646.6	100.0	172,376.1	100.0

cado encontra-se ainda liderado pelo Symbian com uma quota de mercado superior a 37%. O sistema operativo da RIM e iOS da Apple encontram-se, respetivamente, na terceira e quarta posição com 57,5 e 46.6 milhões de unidades vendidas em 2010.

2.2.2 Modelos de negócio

O modelo de negócio baseado em lojas de aplicações tem vindo a ser alvo de bastante interesse [Hammershoj et al., 2010]. Neste modelo de negócio, os utilizadores podem aceder à loja, pagar para adquirir a aplicação, instalar no dispositivo e usufruir das suas funcionalidades. Um exemplo deste modelo de negócio é a loja de aplicações da Apple (App Store). Após o sucesso da loja de aplicações da Apple, vários fabricantes de dispositivos móveis passaram a disponibilizar a sua própria loja de aplicações. Devido a este crescente interesse, é facilmente perceptível que o mercado se encontra bastante fragmentado, com a existência de vários dispositivos, com diferentes sistemas operativos e com plataformas de desenvolvimento incompatíveis entre si, o que torna o desenvolvimento de aplicações um mercado de massa desafiador.

2.2.3 Sistemas operativos móveis

Do ponto de vista técnico, o desenvolvimento de aplicações nativas é simples e consistente. As aplicações nativas são executadas sem a necessidade de um ambiente intermediário. Consequentemente, o acesso aos recursos do dispositivo através da interface de programação disponibilizada pelo sistema operativo é mais rápido, contudo os riscos de segurança aumentam drasticamente [Blom et al., 2008].

A principal questão que se coloca no desenvolvimento de aplicações nativas é o conceito: “*write once, run anywhere*” [Blom et al., 2008]. Como as aplicações nativas são compiladas para um sistema operativo específico, não são tão portáteis como as aplicações desenvolvidas em ambientes mais genéricos. Os sistemas operativos podem utilizar diferentes métodos para alcançar objetivos semelhantes, alguns dos quais podem ser totalmente incompatíveis com outros dispositivos.

Os vários sistemas operativos implementam diferentes formas de interação: ecrãs sensíveis ao toque e com diferentes resoluções, existência de teclas de atalho e diferentes soluções de introdução de dados. Tudo isto condiciona a forma como se desenha uma aplicação. O sucesso de uma aplicação depende de um vasto conjunto de características, entre as quais se encontra o interface com o utilizador. Isto torna quase impossível criar uma aplicação universal.

Este capítulo apresenta uma análise dos principais sistemas operativos para dispositivos móveis. Apenas são considerados os sistemas operativos para dispositivos móveis mais atrativos nos dias de hoje, selecionados com base nos resultados da análise de mercado apresentada anteriormente.

Symbian

O sistema operativo da Nokia é o mais utilizado no mundo dos *smartphones* [Barrera & Oorschot, 2011]. O sistema operativo existe desde o início dos anos 90 e encontra-se, hoje em dia, em inúmeros modelos de *smartphones*, possuindo uma quota de mercado superior a 37%. Começou por ser uma plataforma proprietária, desenvolvida pela Symbian Ltd.³ para, em fevereiro de 2010, passar a ser uma plataforma de software livre.

As ferramentas de desenvolvimento do Symbian permitem produzir aplicações utilizando linguagens de programação como Qt, Symbian C++, Java, Web Runtime (WRT) e Python. No entanto, apesar das várias possibilidades para o desenvolvimento de aplicações, segundo o site oficial, o Qt é atualmente a tecnologia recomendada para a criação de aplicações para Symbian. Como será possível ver mais à frente, a tecnologia Qt disponibiliza aos programadores um conjunto de APIs abrangentes e de fácil utilização para o desenvolvimento de aplicações móveis.

A distribuição das aplicações para dispositivos Nokia baseados no sistema opera-

³ Antes da compra pela Nokia, no final de 2008 a Symbian Ltd. era propriedade da Nokia (56,3%), Ericsson (15,6%), Sony Ericsson (13,1%), Matsushita/Pananonic (10,5%) e Samsung (4,5%).

tivo Symbian, é realizada através da loja de aplicações *online* OVI. Por questões de segurança, o Symbian obriga que todas as aplicações sejam assinadas digitalmente, todavia nem todos os certificados têm de ser emitidos pela Symbian Foundation⁴. Os programadores podem assinar as suas aplicações, permitindo que estas utilizem recursos protegidos como acesso a dados de localização. As aplicações que necessitem de alterar as configurações do sistema, ou acesso a ficheiros protegidos do sistema, devem ser submetidas à aprovação da Symbian Signed⁵ [Barrera & Oorschot, 2011].

Android

Nos últimos anos, surgiram várias plataformas para dispositivos móveis baseadas em Linux, tais como Maemo, Openmoko, Qtopia e LiMo [Oliver, 2009]. Estas plataformas despertaram o interesse de fabricantes como Nokia, Samsung, Motorola, entre outros. No entanto, segundo Oliver [2009], nenhuma delas chamou a atenção e teve o apoio generalizado que o Android da Google obteve.

A Google lançou o Android em meados de 2007, para o desenvolvimento de normas abertas para dispositivos móveis. O Android é uma plataforma de software livre, com uma licença de código aberto para dispositivos móveis baseados em Linux. O Android permite aos programadores desenvolverem software em Java, ou alternativamente em C/C++, utilizando o Native Development Kit (NDK) que contém APIs, documentação e exemplos de aplicações que ajudam o programador a implementar funcionalidades nativas. A arquitetura do Android foi projetada para simplificar a reutilização de componentes, ou seja, qualquer aplicação pode disponibilizar os seus serviços para serem utilizados por qualquer outra aplicação.

O *kit* de desenvolvimento do Android disponibiliza as ferramentas e as APIs necessárias para o acesso às funcionalidades de ecrãs sensíveis ao toque, acelerómetros, gráficos 3D e GPS, bem como a colaboração entre aplicações como *e-mail*, mensagens, agenda, redes sociais e serviços baseados na localização.

As aplicações para o Android podem ser obtidas através da Android Market ou diretamente através das páginas web dos programadores. A Google tem uma intervenção mínima quando as aplicações são submetidas na loja *online*, e nenhuma nas aplicações distribuídas por terceiros. Segundo Barrera & Oorschot [2011], a Goo-

⁴ **Symbian Foundation** é uma organização sem fins lucrativos que realiza toda a administração da plataforma SymbianOS

⁵ **Symbian Signed** é um programa de teste e certificação para aplicações Symbian C++

gle apenas remove as aplicações da sua loja *online*, quando o seu conteúdo viola os termos de utilização ou quando estas são consideradas maliciosas (e mesmo estas, em certas circunstâncias, podem continuar a ser distribuídas). A grande diferença para outros sistemas operativos, é que os programadores de aplicações para Android não necessitam de esperar pela aprovação para que as suas aplicações se encontrem disponíveis.

Blackberry

O Blackberry OS é um sistema operativo proprietário desenvolvido pela Research In Motion (RIM) para os seus dispositivos móveis. A plataforma é utilizada num grande número de modelos, sendo historicamente direccionada para o mercado empresarial devido ao suporte nativo de aplicações que permitem a sincronização com clientes de *e-mail*, calendário, tarefas, notas e contactos.

Segundo Barrera & Oorschot [2011], como plataforma de desenvolvimento o Blackberry possui uma grande robustez e fiabilidade, tem uma panóplia de bibliotecas e permite a interoperabilidade entre as várias aplicações. O suporte à monitorização da bateria e da memória, encontra-se mais refinado do que no sistema operativo Android. A plataforma fornece aos programadores várias APIs para a consulta do estado da bateria, estado do carregamento, temperatura, gestão da luminosidade do ecrã, entre muitos outros.

O Blackberry suporta o desenvolvimento de aplicações desenvolvidas por terceiros. O sistema operativo utiliza um modo seguro para isolar as aplicações em tempo de execução conseguida através da JVM. Tradicionalmente, as aplicações eram distribuídas sem a aprovação da RIM, no entanto, em abril de 2009 foi lançada a Blackberry App World, onde os utilizadores podem aceder a um conjunto de aplicações aprovadas pela RIM através de uma aplicação instalada no dispositivo [Barrera & Oorschot, 2011].

iPhone OS

O iPhone OS (renomeado para iOS em julho de 2010), é o sistema operativo nativo dos dispositivos móveis da multinacional Norte-Americana Apple. Todos os seus dispositivos móveis (iPhone, o iPod Touch e iPad) possuem o iOS, permitindo que os programadores desenvolvam aplicações compatíveis com todos esses dispositivos.

As aplicações do iOS são desenvolvidas em Objective C e são capazes de comunicar com o hardware através de um conjunto de APIs publicadas pela própria Apple. Para que sejam utilizadas APIs registadas, o iOS aplica algumas restrições a aplicações desenvolvidas por terceiros, como por exemplo o acesso a ficheiros do sistema. O iOS disponibiliza várias camadas de abstracção para que facilmente sejam criados menus que interagem com os utilizadores, gráficos 2D e 3D, serviços de localização e funcionalidades do sistema operativo tais como *threads* e *sockets*.

Os programadores que desejam publicar as suas aplicações devem submetê-las à Apple para aprovação [Barrera & Oorschot, 2011]. Apesar de não se encontrarem publicadas as informações detalhadas sobre os critérios de aprovação das aplicações, Barrera & Oorschot [2011] consideram que existe uma verificação manual de pelo menos algumas aplicações. Se a aplicação for classificada como adequada para distribuição pública, é assinada digitalmente pela Apple e lançada na App Store. Contudo, são rejeitadas aplicações que violem a propriedade intelectual ou que vão contra os termos de desenvolvimento de aplicações.

Windows Phone 7

Windows Phone 7 ou apenas WP7, é um sistema operativo proprietário desenvolvido pela Microsoft e sucessor da plataforma Windows Mobile. É baseado no Kernel Windows CE 6.0 R3, no entanto também já utiliza alguns dos recursos do Windows Embedded Compact 7. Este consiste num sistema operativo de tempo real desenvolvido para suportar várias arquiteturas de CPU. Além de suportar processadores com dois núcleos, o Windows Embedded Compact 7 possui características como Bluetooth 2.1 EDR, Digital Living Network Alliance (DLNA)⁶, Media Transfer Protocol (MTP)⁷, entre muitas outras.

Segundo o site oficial, o desenvolvimento para WP7 ocorre utilizando um conjunto de ferramentas que a Microsoft disponibiliza gratuitamente. O *kit* encontra-se disponível em vários idiomas e inclui o seguinte:

- Visual Studio 2010 Express for Windows Phone
- Windows Phone Emulator Resources

⁶ Digital Living Network Alliance (DLNA) é uma norma desenvolvida por várias empresas de tecnologia com o objetivo de simplificar a comunicação entre vários dispositivos.

⁷ Media Transfer Protocol (MTP) é um protocolo desenvolvido pela Microsoft para realizar a ligação entre portáteis e computadores com sistema operativo Windows.

- Silverlight 4 Tools For Visual Studio
- XNA Game Studio 4.0
- Microsoft Expression Blend for Windows Phone

Segundo o site oficial da Microsoft, estas ferramentas permitem a criação, codificação, testes e ainda o tratamento do aspeto gráfico da aplicação com a ferramenta “Microsoft Expression Blend”. Além disso, permite a utilização de duas plataformas para o desenvolvimento de aplicações:

- Silverlight - Suporta a criação de aplicações em C# ou Visual Basic (VB) e encontra-se direcionada para a criação de *Rich Internet Applications*.
- XNA Framework - Vocacionada para a criação de jogos para a plataforma *gaming* da Microsoft.

As aplicações desenvolvidas nas plataformas Silverlight e XNA Framework são, hoje em dia, compatíveis com praticamente todos os dispositivos móveis com Windows Phone 7, sendo apenas necessários pequenos ajustes como tamanho do ecrã ou recursos específicos do dispositivo.

Quando a aplicação se encontrar completa, o programador pode disponibilizar a aplicação na Windows Phone Marketplace (loja de aplicações *online* da Microsoft). Todas as aplicações, gratuitas ou comerciais, apenas poderão ser adquiridas através deste meio de distribuição. De salientar que as aplicações desenvolvidas são submetidas a um processo de certificação, de modo a verificar as políticas da Microsoft que regulam a parte técnica e de conteúdo das aplicações móveis.

2.2.4 Outras plataformas de desenvolvimento

Durante a presente subsecção foram analisados vários sistemas operativos, com os correspondentes modelos de negócio, com as suas próprias ferramentas de desenvolvimento e com diferentes formas de distribuição das aplicações. Deduz-se facilmente que o mercado se encontra bastante fragmentado, com uma quantidade considerável de soluções distintas e incompatíveis entre si, tornando difícil a utilização das aplicações em diferentes dispositivos móveis.

Na sequência da análise realizada anteriormente, apresenta-se de seguida uma pequena revisão de algumas tecnologias/plataformas que, de alguma forma, procuram

aumentar a portabilidade das aplicações entre os diferentes sistemas, e libertar os programadores dos detalhes específicos de cada sistema operativo.

Java ME

Java Platform Micro Edition (Java ME), ou J2ME, é uma tecnologia que possibilita o desenvolvimento de software para dispositivos móveis. A plataforma Java ME é executada numa máquina virtual e permite um razoável, mas não completo, acesso às funcionalidades básicas dos dispositivos. As aplicações baseadas em Java ME são compatíveis com vários dispositivos, acrescentando dessa forma recursos às funcionalidades nativas de cada dispositivo [Gavalas & Economou, 2011].

Java ME é um conjunto de APIs Java definidas pela Java Community Process (JCP). A JCP destina-se a um processo de especificação, permitindo que qualquer utilizador interessado se possa envolver na especificação das versões da plataforma Java. O processo JCP utiliza os documentos Java Specification Requests (JSRs) que descrevem não só as especificações propostas mas também as tecnologias que se pretendem acrescentar à plataforma Java. A comunidade JCP adotou duas abordagens para especificar as necessidades dos pequenos dispositivos [Oracle Corporation, 2011]:

- Connected Limited Device Configuration (CLDC): configuração que define uma norma na plataforma Java, para os dispositivos com recursos limitados, ativando de forma dinâmica a distribuição das aplicações e conteúdos para estes dispositivos.
- Mobile Information Device Profile (MIDP): conjunto de bibliotecas Java que, juntamente com a configuração CLDC, fornece os requisitos mínimos para a construção de uma aplicação J2ME, direccionada para telefones móveis, PDAs e *two-way pagers*.

Java ME é a plataforma de software móvel dominante. No entanto, o conceito da linguagem Java “*write once, run anywhere*”, não se aplica integralmente no Java ME. Devido às diferentes formas de interação que existem nos vários sistemas operativos, os programadores devem disponibilizar várias versões da aplicação de modo a lidar com a ampla gama de dispositivos. No entanto Gavalas & Economou [2011], afirmam que desenvolvendo uma aplicação em Java ME tendo em vista apenas um sistema operativo, é possível atingir cerca de 70% dos dispositivos com esse sistema operativo.

Microsoft .NET CF

O Microsoft .NET⁸ (popularmente conhecido por .NET Framework) é uma iniciativa da empresa Microsoft, que visa uma plataforma única para desenvolvimento e execução de aplicações. O .NET para dispositivos móveis, designado por .NET Compact Framework (.NET CF), utiliza algumas das bibliotecas que o .NET Framework disponibiliza para o desenvolvimento de aplicações em computadores pessoais e ainda algumas bibliotecas desenvolvidas exclusivamente para dispositivos móveis [Gavalas & Economou, 2011].

O princípio do .NET CF é análogo à Java Virtual Machine (JVM). Inicialmente, a Microsoft desenvolveu a plataforma .NET com suporte para vários sistemas operativos, procurando alcançar uma grande quantidade de programadores e reutilizar as bibliotecas existentes. No entanto, atualmente a ferramenta de desenvolvimento do .NET (o Visual Studio) suporta apenas C# e Visual Basic. Além disso, restringe a sua utilização a plataformas Windows.

Segundo Gavalas & Economou [2011], o .NET CF disponibiliza uma integração satisfatória com as funcionalidades específicas de comunicação do dispositivo como serviço de voz, SMS e Bluetooth. Contudo, tem como alvo um conjunto muito limitado de dispositivos e as ferramentas de desenvolvimento incluem custos de licenças.

Python

De acordo com Hammershoj et al. [2010], Python é uma excelente linguagem de programação para a realização de protótipos. Além de ser uma linguagem de fácil aprendizagem é também de fácil programação. Dependendo do sistema operativo, existem diferentes versões da linguagem Python. Por exemplo, para o sistema operativo Symbian existe a versão PyS60. Os programas desenvolvidos em Python são, normalmente, mais curtos do que os programas equivalentes em linguagens de programação como C, C++ ou mesmo Java, devido às seguintes características:

- Os tipos de dados de alto nível permitem expressar operações complexas numa única operação;
- O agrupamento de instruções é realizado através de indentação em vez da utilização de parênteses de início e de fim;

⁸ Em Inglês lê-se dotNet.

- Não é necessário declarar variáveis nem argumentos.

A versão PyS60 é basicamente uma extensão do Python para o sistema operativo Symbian (S60). As extensões ao Python encontram-se divididas em dois módulos: e32 e appuifw. O primeiro módulo (e32), é responsável pela manipulação de objetos e funções nativas do sistema operativo Symbian, enquanto que o segundo, appuifw, contém os elementos da plataforma S60 necessários para a interface gráfica com o utilizador.

Segundo Hammershoj et al. [2010], PyS60 é uma boa linguagem de programação para a realização de protótipos. Este facto, deve-se não só à inexistência de restrições no acesso a funcionalidades locais, como também à abundância de APIs que permitem, por exemplo, o acesso ao sistema de ficheiros, a funcionalidades da câmara fotográfica, a interfaces de comunicação, entre outras.

A necessidade de instalação de um interpretador é uma das grandes desvantagens da utilização de Python. Devido a essa necessidade, a velocidade de processamento é bastante afetada, não sendo por isso direcionada para aplicações de alto desempenho. Além disso, sistemas operativos como iOS, Android, Blackberry e Windows Mobile, não suportam oficialmente a linguagem Python.

Adobe AIR

Adobe AIR [Adobe Inc., 2011] é uma solução multiplataforma que possibilita o desenvolvimento de aplicações avançadas. Atualmente encontra-se na versão 2.5, suportando várias APIs para, por exemplo, acesso às funcionalidades de ecrãs *multi-touch*, acelerómetros, câmaras, entre outras.

Um dos principais objetivos desta plataforma de desenvolvimento, é permitir a reutilização de código existente para o desenvolvimento de aplicações, tendo em vista vários dispositivos e várias plataformas. O Adobe AIR 2.5 é suportado por sistemas operativos como BlackBerry, Android, iOS, Windows e Linux.

O desenvolvimento de aplicações com o Adobe AIR pode efetuar-se em ferramentas como Adobe Flash e Flex Builder. Já se encontram disponíveis, em lojas de aplicações *online* como Android Market e App Store, inúmeras aplicações desenvolvidas com o Adobe AIR. Além disso, a empresa Adobe anunciou, em outubro de 2010, o Adobe InMarket. Este serviço permite aos programadores a distribuição das suas aplicações, para diferentes tipos de dispositivos, em lojas de aplicações da Acer, Intel, etc.

Segundo David Wadhwani⁹, “Com o lançamento do AIR 2.5, mais de 3 milhões de programadores Flash podem agora construir um único jogo ou aplicação e facilmente colocar em lojas de aplicações de vários tipos de dispositivos”. Para comprovar esta afirmação, Christian Cantrell da Adobe, criou um jogo que funciona em cinco plataformas sem a necessidade de alterar uma única linha de código. No vídeo que se encontra na notícia¹⁰ colocada no blog Engadget, é possível ver o jogo a funcionar perfeitamente em Mac OS X, Windows 7, Ubuntu Linux, iPhone, Android e iPad [Hollister, 2011].

Qt

Qt é um *framework* multiplataforma que pode ser utilizado em sistemas operativos como Windows, Mac e Linux. Em dispositivos móveis é apenas suportado por Symbian e Maemo. Foi originalmente criado pela empresa Norueguesa Trolltech no entanto, em junho de 2008, foi adquirido pela Nokia.

O Qt disponibiliza um considerável conjunto de blocos para o desenvolvimento de aplicações em C++. De acordo com Hammershoj et al. [2010], em relação à comunicação entre objetos e à flexibilidade para o desenvolvimento de interfaces gráficas avançadas, o Qt consegue ser superior a linguagens como C++.

Desde 2005, o Qt teve uma rápida fase de desenvolvimento fazendo com que hoje, seja uma das mais importantes linguagens de programação para dispositivos móveis. A versão mais recente inclui o QtCreator que permite um desenvolvimento rápido de aplicações.

Enquanto que o conceito do Java é “*write once, run everywhere*”, o do Qt é “*write once, compile everywhere*” [Hammershoj et al., 2010]. Em comparação com o Java, tem a vantagem de ser compilado diretamente para o sistema operativo, não necessitando portanto de nenhuma camada adicional como JVM.

⁹ David Wadhwani: Vice-presidente da unidade de negócios de soluções criativas/interativas da Adobe.

¹⁰<http://engadget.com/2010/04/05/adobe-air-developer-demonstration-one-game-five-platforms-all/>.

2.3 Soluções propostas anteriormente

A evolução tecnológica que se tem verificado nos dispositivos móveis e nas tecnologias que incorporam, faz com que surjam constantemente inúmeras aplicações. Facilmente se encontram aplicações que, de alguma forma, procuram tirar proveito da localização dos dispositivos. No entanto, não é conhecida qualquer aplicação que se relacione diretamente com a proposta apresentada neste trabalho. A proximidade entre os dispositivos móveis é, na maior parte das vezes, utilizada para questões relacionadas com publicidade ou partilha de informação, não é no entanto utilizada para a colaboração entre os dispositivos de modo a atingirem um objetivo comum.

2.4 Conclusões

Pelo que se apresenta neste capítulo, conclui-se que existem diversos sistemas operativos, alguns bastante diferentes no funcionamento e interação com o utilizador. Existem inúmeras linguagens e plataformas de programação. Algumas procuram ser genéricas ao ponto de permitir que a aplicação funcione em diferentes sistemas operativos e dispositivos.

A realidade faz ver que todas as linguagens/plataformas são utilizadas. Seja porque existem programadores que vão continuar a dar preferência a linguagens que já conhecem, seja porque algumas aplicações apenas podem ser construídas numa determinada linguagem.

As inúmeras soluções existentes no mercado, fazem com que não seja possível estabelecer uma solução universal.

Capítulo 3

Tecnologias de comunicações sem fios

Na atualidade, as tecnologias de comunicações sem fios têm sido alvo de um desenvolvimento muito rápido, dando origem a diversas áreas de aplicação. Várias normas e tecnologias de redes sem fios surgiram nos últimos anos. Foram desenvolvidas desde redes móveis celulares, a redes de dados sem fios e a redes Personal Area Networks (PANs), utilizadas na comunicação entre equipamentos pessoais. Estas tecnologias, encontram-se cada vez mais presentes em dispositivos dos mais variados tipos, possibilitando a comunicação com outras pessoas e, ao mesmo tempo, facilitando a obtenção de conhecimento. Devido à mobilidade que permitem, estas redes tornaram possível a obtenção de informação em qualquer lugar e a qualquer momento.

Este capítulo descreve algumas das tecnologias de comunicação sem fios que se encontram disponíveis nos dias de hoje. Embora grande parte do capítulo seja focado na norma IEEE 802.11 e na tecnologia Bluetooth, é importante verificar se alguma das restantes tecnologias se apresenta como uma melhor solução para a resolução do problema.

3.1 IrDA - Infrared Data Association

Infrared Data Association (IrDA) é uma organização que define normas para a utilização de sistemas de comunicação sem fios, utilizando a tecnologia dos infravermelhos. A especificação IrDA refere-se essencialmente a ligações ponto-a-ponto, onde exista linha de vista entre os intervenientes. É um tipo de interface bastante

económica, encontrada em grande parte dos dispositivos de computação móvel (telefones móveis, portáteis, impressoras, etc.) mais antigos, suportando velocidades de comunicação até cerca de 16 Mbps [Vitsas & Barker, 2003].

Recentemente a IrDA adotou um novo protocolo (IrSimple), com o objetivo de disponibilizar comunicações sem fios simples e com velocidades de transmissão elevadas, entre dispositivos móveis e eletrodomésticos digitais. Segundo o site oficial, o IrSimple permite que dispositivos móveis (como telefones móveis e máquinas fotográficas) transmitam imagens para monitores ou projetores para uma visualização imediata.

A curta distância necessária (mesmo nas versões mais recentes do protocolo) e a necessidade de linha de vista entre os intervenientes envolvidos na comunicação, torna a tecnologia bastante inadequada para a disseminação de mensagens para um grande grupo de dispositivos.

3.2 RFID - Radio Frequency Identification

Radio Frequency IDentification (RFID) é uma especificação para a identificação de objetos com *tags* RFID¹. Tipicamente as *tags* RFID não suportam mais do que 2 KB de informação, restringindo desse modo, a quantidade de informação que podem transmitir. O alcance de comunicação varia entre apenas alguns milímetros até vários metros utilizando *tags* passivas², até mais de 100 metros utilizando *tags* ativas³. O alcance depende ainda de fatores como a qualidade do recetor RFID utilizado, e da interferência existente no ambiente [RFID Journal, 2011].

Atualmente, os sistemas RFID encontram-se presentes em todos os lados sem se dar conta da sua existência [RFID Journal, 2011]. Esta tecnologia têm-se tornado parte integrante do quotidiano das pessoas encontrando-se presente em diversas áreas. Alguns exemplos de aplicações são: pagamento de portagens em movimento, controlo de entradas em edifícios para pessoas e automóveis, monitorização de produtos em armazéns e lojas, entre outros. Ainda assim poucos telefones móveis suportam RFID.

¹ *Tags* RFID são pequenos *chips* com antenas que contêm informação, ou apenas um código de identificação, do objeto a que se encontram associadas (<http://pt.wikipedia.org/wiki/RFID>).

² *Tags* sem a sua própria fonte de energia.

³ *Tags* que possuem a sua própria fonte de energia.

3.3 NFC - Near-Field Communication

Near Field Communication (NFC) é uma tecnologia em crescimento que concentra a sua atenção em comunicações sem fios de curto alcance (apenas alguns centímetros). Em meados de 2004, um grupo de empresas fundou o NFC Forum, com o objetivo de promover a implementação da tecnologia NFC, de forma a garantir uma maior interoperabilidade entre um grande número de dispositivos e serviços. O NFC é uma combinação de tecnologias de comunicação com o RFID. Esta tecnologia permite uma comunicação bidirecional através da aproximação dos intervenientes [Paus, 2007].

Começando por ser uma tecnologia com um elevado custo e com pouca penetração no mercado encontra-se em constante desenvolvimento. O número de aplicações que utilizam o NFC tem aumentado esperando-se um futuro promissor. Os *smartphones* são os dispositivos móveis de eleição para a implementação desta tecnologia [Want, 2011]. A Google disponibilizou esta tecnologia no telemóvel Nexus S, desenvolvido em parceria com a Samsung e lançado em dezembro de 2010. As áreas onde existem mais aplicações são: pagamentos, chaves eletrónicas, identificação e receção/envio de informação [Paus, 2007].

A tecnologia NFC funciona mundialmente na banda de frequências Industrial-Scientific-Medical (ISM)⁴ dos 13.56 MHz. Atualmente, disponibiliza velocidades de transferência de dados de 106, 212 e 424 kbps sendo no futuro, esperadas velocidades superiores. Apesar da tecnologia se encontrar projetada para comunicações até uma distância máxima de 20 centímetros, é normalmente utilizada em menos de 10 cm [Paus, 2007]. Assim sendo, uma vez que a tecnologia NFC se encontra direcionada para a transmissão de dados entre dispositivos que se encontrem a poucos centímetros de distância, é inadequada para o envio de mensagens para um grande grupo de dispositivos.

⁴ ISM é uma gama de frequências que se encontram reservadas para aplicações industriais científicas ou médicas. As bandas ISM não necessitam de licenciamento. No entanto, não existe um consenso a nível mundial sobre as bandas de funcionamento.

3.4 ZigBee

O ZigBee é uma tecnologia de comunicações sem fios baseada na norma IEEE 802.15.4. A entidade responsável pelo ZigBee intitula-se por ZigBee Alliance⁵. O ZigBee pretende disponibilizar uma transmissão de dados sem fios com um baixo consumo energético e com elevada fiabilidade [ZigBee Alliance, 2011].

O ZigBee funciona nas frequências: 2.4GHz (Global), 915Mhz (na América) e 868Mhz (na Europa) com velocidades de 250, 40 e 20 kbps respetivamente. Dependendo do ambiente de utilização e do consumo de energia de cada aplicação, o alcance de comunicação do ZigBee varia entre 10 e 1600 metros [ZigBee Alliance, 2011].

Esta tecnologia não é direcionada para a transferência de grandes quantidades de informação. Além disso, os dispositivos que implementam esta tecnologia, são principalmente utilizados para fins de monitorização, automação e controlo. Tal como a tecnologia RFID, os dispositivos móveis que normalmente acompanham os utilizadores, não implementam esta tecnologia. Desta forma, o ZigBee não é relevante no contexto do presente trabalho.

3.5 Rede móvel celular

A primeira geração de redes móveis (1G) apareceu pela primeira vez no início dos anos 80. Essa primeira geração era baseada em técnicas de modulação analógica, que apenas permitia a transmissão de voz. Os sistemas móveis de primeira geração não suportavam encriptação da informação (problemas de segurança), a qualidade do serviço de voz era fraca e a velocidade de transferência era de aproximadamente 9.6 Kbps [Li et al., 2009].

Os sistemas de segunda geração (2G) apareceram por volta de 1990 no sentido de corrigir as limitações dos sistemas móveis de comunicação de primeira geração [Li et al., 2009]. Além da maior capacidade disponibilizada pelos sistemas digitais, estes proporcionavam uma melhor qualidade de serviço, uma vez que não se encontram tão sujeitos a ruídos e interferências. Existem vários sistemas de segunda geração sendo o mais popular o GSM.

Além do transporte de voz, as redes móveis de segunda geração possibilitavam a

⁵ A ZigBee Alliance é uma associação sem fins lucrativos formada por membros provenientes de várias partes do mundo e de vários tipos de organizações (empresas multinacionais, empresas publicas, universidades, etc.).

transmissão de dados, porém com débitos muito reduzidos (14.4 kbps) [De Vriendt et al., 2002]. Com o desenvolvimento de novos serviços e com o aumento do tráfego, surgiram extensões⁶ (como por exemplo: High Speed Circuit-Switched Data (HSCSD) e GPRS) aos sistemas de segunda geração que suportavam velocidades de transmissão de dados superiores.

A terceira geração de redes móveis (3G), caracteriza-se pelo suporte a serviços de transmissão de dados sem fios com velocidades superiores [Pinto, 2011]. Estes sistemas permitiram às operadoras de rede proporcionar aos seus utilizadores uma quantidade considerável de serviços, que vão desde o suporte a aplicações multimédia (vídeo, áudio, dados), ao acesso a vários serviços disponíveis na Internet (World Wide Web (WWW), *e-mail*, etc.).

A quarta geração de redes móveis (4G), aparece de forma a melhorar os sistemas de terceira geração [Pinto, 2011]. Com esta geração, espera-se desde logo um conjunto de vantagens que aproximará a experiência de utilização dos serviços móveis à das comunicações fixas: grandes velocidades de transmissão de dados, grande cobertura e boa qualidade de serviço. Tecnologias como o WiMax ou LTE, introduzidas no mercado em 2006, têm sido designadas de tecnologias 4G.

A rede móvel suporta várias formas de difusão de mensagens. Uma das estratégias mais vulgares é a utilização dos serviços SMS e Multimedia Messaging Service (MMS). A grande vantagem da utilização desses serviços é que, presentemente, são suportados pela generalidade de telefones móveis. A desvantagem para a utilização desse serviços é o custo associado à sua utilização (em especial quando os utilizadores são clientes de diferentes operadoras). Além disso, estes serviços requerem uma lista de utilizadores previamente conhecidos, para ser possível enviar a mensagem [Ho, 2006].

3.6 IEEE 802.11

A norma IEEE 802.11 aparece, no início dos anos 90, como uma especificação para criação de redes sem fios, em tudo equivalente às convencionais redes de área local com fios (Local Area Networks (LANs)), como por exemplo Ethernet.

A norma IEEE 802.11 define as especificações para as camadas física e de acesso ao meio. Esta norma, para redes sem fios de área local (WLAN), foi originalmente

⁶ Estas extensões denominam-se por sistemas 2.5G uma vez que servem de transição para os sistemas de terceira geração.

publicada em 1997 com velocidade de transmissão de dados até 2 Mbps. Desde então, foram criados vários grupos de trabalho (designados pelas letras “a”, “b”, “c”, etc.) que têm como objetivo fornecer melhorias como qualidade de serviço e novas funcionalidades à norma IEEE 802.11.

Os grupos de trabalho 802.11a e 802.11b, disponibilizaram duas evoluções relevantes à norma original⁷. O IEEE 802.11a define uma nova camada física que utiliza frequências na ordem dos 5 GHz com modulação Orthogonal Frequency-Division Multiplexing (OFDM). Esta evolução permite as seguintes velocidades de transmissão de dados: 6, 9, 12, 18, 24, 36, 48 e 54 Mbps. O IEEE 802.11b, especifica também uma nova camada física, e permite atingir velocidades de transmissão até 11 Mbps com um alcance que pode ir até 300 metros em espaço aberto. Esta evolução, utiliza a camada física que opera na banda de frequências ISM dos 2.4 GHz e, o acesso ao meio partilhado pelos dispositivos, é baseado numa técnica de espalhamento espectral por sequência direta (Direct Sequence Spread Spectrum (DSSS)).

Entre os restantes grupos de trabalho, vale a pena destacar o grupo de trabalho 802.11e, que tem vindo a reforçar a camada Medium Access Control (MAC) com recursos de Quality of Service (QoS), para suportar voz e vídeo sobre redes 802.11, e o grupo de trabalho 802.11n, que tem vindo a trabalhar na evolução do 802.11g de forma a fornecer velocidades de transmissão de dados superiores, permitindo transmissão de dados multimédia, Voice Over Internet Protocol (VoIP), Internet Protocol Television (IPTV), entre outros.

3.6.1 Arquitetura do sistema

A arquitetura da norma IEEE 802.11 (figura 3.1) consiste em vários componentes que, em conjunto, permitem a criação de uma rede local sem fios, suportando a mobilidade de estações de modo transparente para as camadas superiores.

A norma IEEE 802.11 define os modos de funcionamento Basic Service Set (BSS) e Independent Basic Service Set (IBSS), que correspondem respetivamente à rede baseada em infraestrutura e à rede *ad-hoc*.

De acordo com a especificação [IEEE 802.11, 2007], um agrupamento de duas ou mais estações, sujeito ao controlo direto da mesma função de coordenação, constitui uma BSS. A área coberta por uma BSS é denominada de Basic Service Area (BSA).

⁷ As evoluções disponibilizadas pelos grupos de trabalho 802.11a e 802.11b são vulgarmente referidas como WiFi.

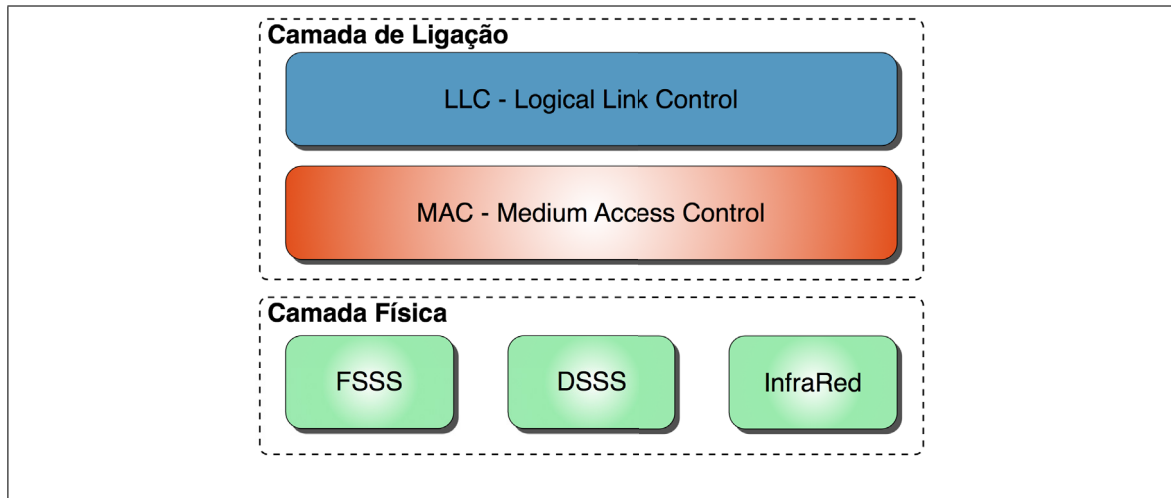


Figura 3.1: Camadas da especificação IEEE 802.11. [Schiller, 2003].

Estas redes, encontram-se configuradas de forma a que os nós não possam comunicar diretamente entre si, realizando a comunicação apenas com os pontos de acesso que servem de interface entre o segmento sem fios e o resto da rede. Os pontos de acesso não só proporcionam o acesso a outras redes, como também possuem funções de encaminhamento, controlo de acesso ao meio, etc. Como o ponto de acesso é o elemento responsável pelo controlo de acesso ao meio, possui autoridade para regular o acesso por parte das estações. Desta forma, uma estação (STA) que deseje transmitir, deve aguardar a autorização do ponto de acesso. A figura 3.2 ilustra um exemplo de uma rede IEEE 802.11 baseada em infraestrutura.

Como se verifica na figura 3.2, é possível ligar vários pontos de acesso através de uma ligação designada por Distribution System (DS). Desta forma, os pontos de acesso de diferentes BSSs podem comunicar entre si, proporcionando uma maior mobilidade para as estações. A especificação IEEE 802.11 não define o sistema DS, podendo este ser baseado em diversos tipos de tecnologias (como Ethernet).

O segundo tipo de configuração definida pela norma IEEE 802.11 é denominado de IBSS⁸ (figura 3.3). Uma IBSS é uma rede independente, onde cada estação comunica diretamente com outras estações, sem a necessidade da existência de um ponto de acesso. As estações podem comunicar quando se encontram fisicamente próximas, ou seja, quando se encontram ao alcance de outras estações.

⁸ Vulgarmente conhecido como rede *ad-hoc*.

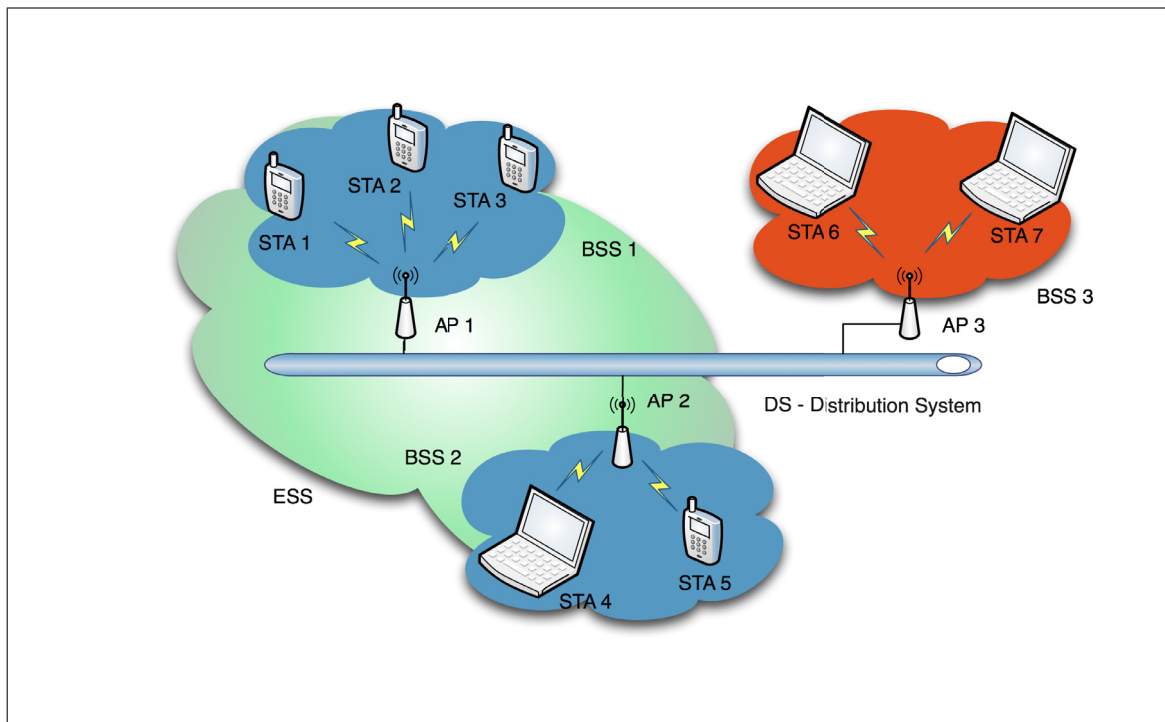


Figura 3.2: Arquitetura de uma rede IEEE 802.11 baseada em infraestrutura [Schiller, 2003].

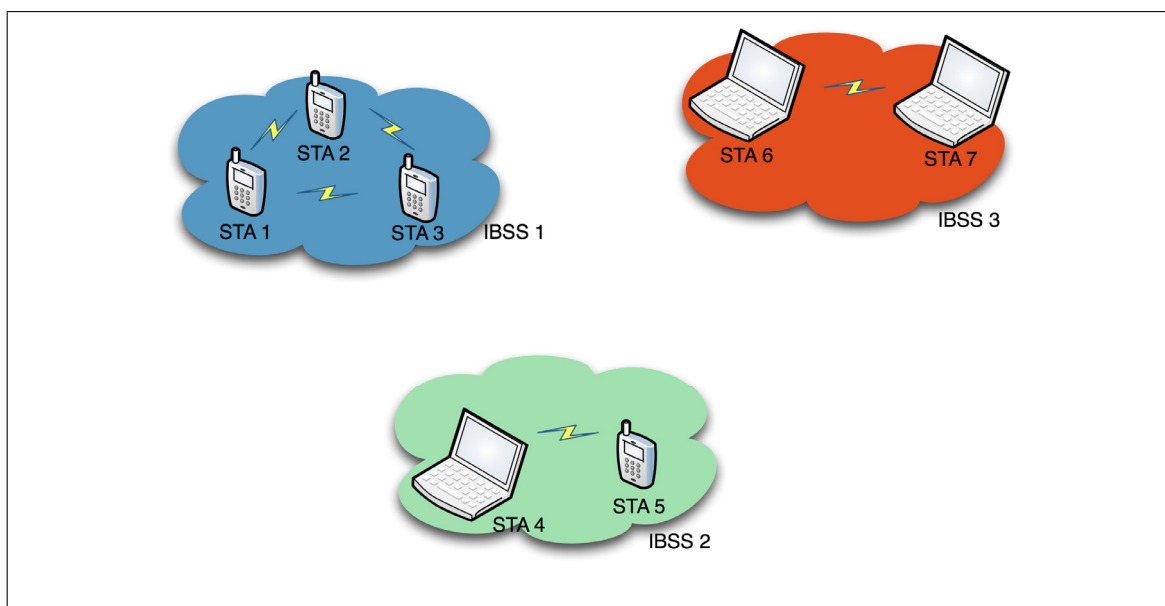


Figura 3.3: Arquitetura de uma rede *ad-hoc* IEEE 802.11 [Schiller, 2003].

3.6.2 Camada física

O IEEE 802.11 define três tipos de camada física [IEEE 802.11, 2007]: Infra Red (IR), Frequency Hopping Spread Spectrum (FHSS) e DSSS. A norma especifica o suporte a velocidades de transmissão de 1 e 2 Mbps. No entanto, surgiram grupos de trabalho, como o 802.11a e 802.11b, com o objetivo de aumentar essas velocidades. De seguida descrevem-se as camadas físicas definidas na especificação [Schiller, 2003]:

- **IR** - Esta camada física caracteriza-se pela transmissão através de infravermelhos, utilizando um sinal ótico com comprimento de onda entre os 850 e os 950 nm. A especificação, prevê a comunicação entre duas estações sem que exista um linha-de-vista entre elas e separadas por distâncias até 10 metros (em locais onde não existam interferências à difusão). O ambiente de utilização característico é no interior de edifícios (salas de aulas, salas de reuniões, etc.).
- **FHSS** - Esta técnica de espalhamento espectral permite a coexistência de várias redes na mesma área, realizando a separação das redes com diferentes sequências de saltos de frequência. Na Europa é respeitada a normalização European Telecommunications Standards Institute (ETSI) de sequências de 26 saltos (distribuídos por 79 canais de 1 MHz). Cada um dos canais é utilizado durante um período de tempo, comutando de seguida para outro canal e assim sucessivamente durante toda a transmissão. A sequência é pseudo-aleatória, o que torna o canal seguro uma vez que a sequência é apenas conhecida pelo emissor e pelo recetor.
- **DSSS** - Na técnica de espalhamento espectral de sequência direta, a separação dos canais é alcançada através da transformação do sinal. A cada bit do sinal a transmitir é aplicado um código conhecido como sequência de Barker. A principal característica deste mecanismo é a robustez contra interferências. No entanto, a implementação é mais complexa do que o mecanismo FHSS.

3.6.3 Camada MAC

A camada MAC (figura 3.4) encontra-se incumbida de várias tarefas. Além da responsabilidade de controlar o acesso ao meio, deve disponibilizar suporte para *roaming*⁹, autenticação e gestão de energia [Schiller, 2003].

⁹ Quando uma estação se encontra em movimento e muda de ponto de acesso.

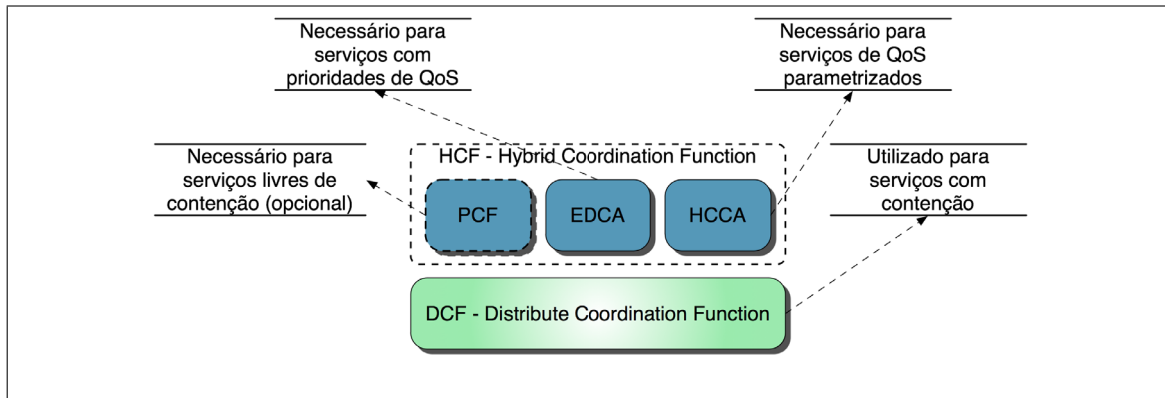


Figura 3.4: Camada de controlo de acesso ao meio do IEEE 802.11 [IEEE 802.11, 2007].

A especificação da camada MAC do IEEE 802.11, define quatro mecanismos de controlo de acesso ao meio [IEEE 802.11, 2007]: um mecanismo básico e obrigatório baseado numa versão do Carrier Sense Multiple Access/Collision Avoidance (CS-MA/CA); um mecanismo opcional que procura evitar o problema da estação oculta; um mecanismo livre de contenção e um mecanismo adicional utilizado apenas em redes com requisitos especiais de QoS. Os dois primeiros mecanismos são denominados de função de coordenação distribuída (Distributed Coordination Function (DCF)), o terceiro de função de coordenação pontual (Point Coordination Function (PCF)) e o último é denominado de função de coordenação híbrida (Hybrid Coordination Function (HCF)).

DCF

O DCF é o mecanismo básico de controlo de acesso ao meio do IEEE 802.11. É baseado no mecanismo de acesso aleatório CSMA/CA. No CSMA as estações “escutam” o meio de forma a detetar se está livre. Quando uma estação deteta que o meio se encontra livre a transmissão ocorre de imediato. No entanto, se duas estações verificam (simultaneamente) a disponibilidade do meio, pode ocorrer uma colisão. De forma a diminuir a probabilidade da existência de colisões, a norma 802.11 especifica um mecanismo que procura evitar a ocorrência de colisões, ou seja, uma estação que deseje transmitir, encontra-se obrigada a verificar o meio durante um tempo aleatório antes de iniciar a transmissão.

Na figura 3.5 é representado o mecanismo básico de controlo de acesso ao meio. Sinteticamente, se o meio se encontra livre há pelo menos um intervalo de tempo DIFS, a estação transmite de imediato. Caso contrário, de forma a atrasar a tentativa

de acesso ao meio, cada estação escolhe um tempo aleatório de *backoff*. Se até ao final do tempo de *backoff* a estação encontrar o meio livre, pode transmitir. Contudo, em caso contrário aguarda por uma próxima oportunidade, ou seja, até que o meio se encontre livre por pelo menos DIFS [IEEE 802.11, 2007].

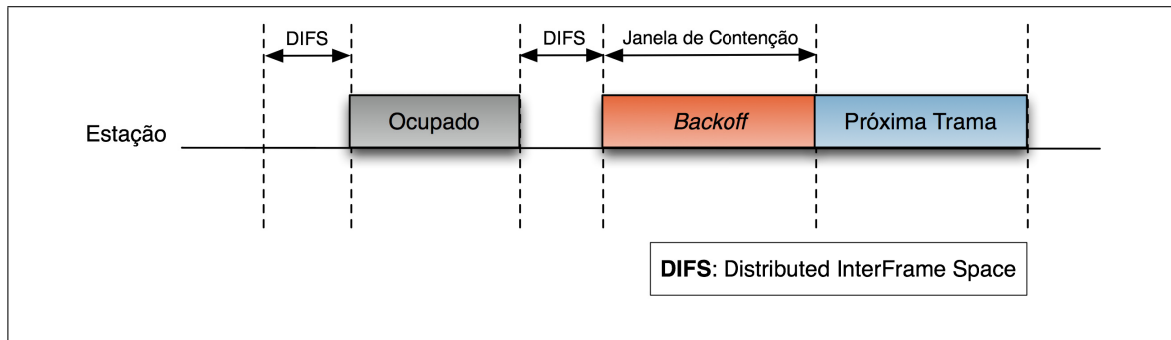


Figura 3.5: Mecanismo básico de controlo de acesso ao meio do IEEE 802.11 [IEEE 802.11, 2007].

RTS/CTS

Para lidar com o problema da estação oculta, o protocolo de controlo de acesso ao meio do IEEE 802.11 define um mecanismo opcional baseado na troca de duas mensagens de controlo: Request To Send (RTS) e Clear To Send (CTS). A figura 3.6 representa a utilização do mecanismo RTS/CTS. Depois de aguardar um intervalo de tempo DIFS (mais um intervalo de *backoff* se o meio estava a ocupado), o emissor pode enviar uma mensagem RTS de forma a solicitar o envio. No caso do recetor da mensagem de dados receber uma mensagem RTS responde, após um intervalo de SIFS, com uma mensagem CTS. No caso da trama CTS não ser recebida corretamente, a estação de origem inicia o processo de *backoff* para a retransmissão da trama RTS. Por outro lado, se a troca das mensagens de controlo RTS e CTS for realizada com sucesso, a estação pode proceder à transmissão da informação desejada, após um período de tempo SIFS.

Da mesma forma que podem ocorrer colisões entre mensagens de dados, podem também ocorrer colisões entre as tramas de controlo. Contudo, como estas são relativamente pequenas, o tempo perdido é menor originando um aumento da eficiência do mecanismo de controlo de acesso ao meio. Devido ao *overhead* associado às mensagens de controlo, a utilização do mecanismo nem sempre é justificada, especialmente na transmissão de mensagens de dados de pequena dimensão [IEEE 802.11, 2007].

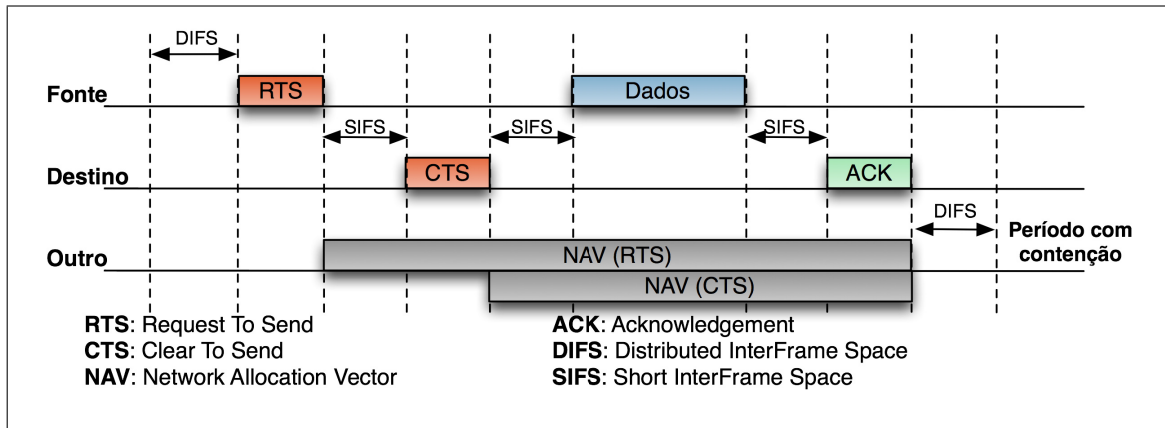


Figura 3.6: Mecanismo RTS/CTS do IEEE 802.11 [IEEE 802.11, 2007].

PCF

O PCF é um mecanismo opcional [IEEE 802.11, 2007]. Devido ao facto de se encontrar baseado num mecanismo de *polling*, permite que as estações utilizem o meio sem contenção. A utilização deste mecanismo é conveniente no transporte de informação sensível a atrasos.

Como se ilustra na figura 3.7, o período livre de contenção é iniciado através da transmissão de uma trama *beacon*. De seguida, o ponto de acesso envia uma mensagem de controlo à primeira estação que se encontra na lista de *polling*. Após um período de SIFS, a estação pode iniciar a transmissão de dados. Como se verifica através da análise da figura 3.7, no caso da estação não iniciar a transmissão de dados, após um período de PIFS o ponto de acesso envia uma nova mensagem de *polling*.

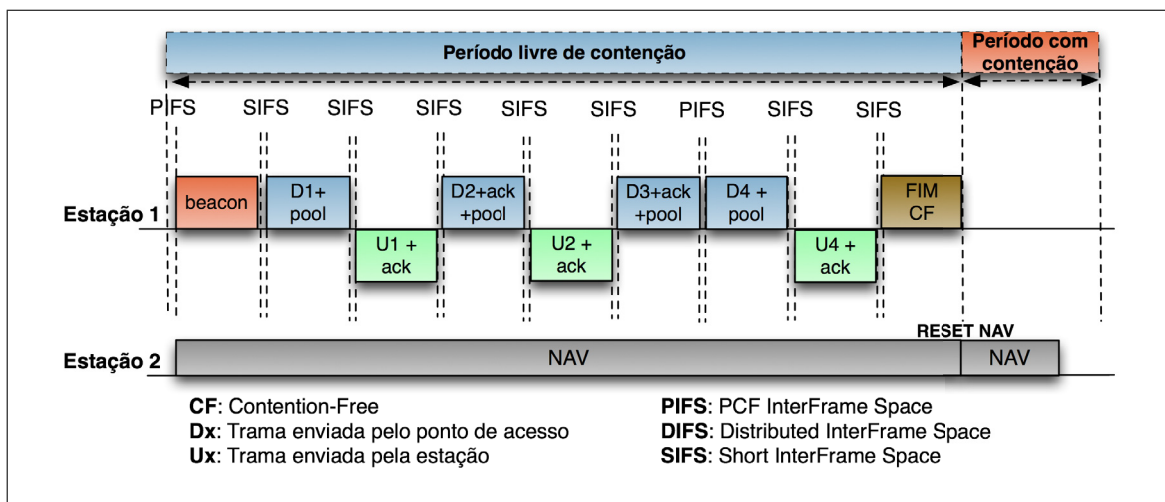


Figura 3.7: Mecanismo de acesso ao meio livre de contenção (PCF) [IEEE 802.11, 2007].

3.7 Bluetooth

O Bluetooth é uma tecnologia de comunicações sem fios de curto alcance (máximo de 100 metros). O objetivo inicial desta tecnologia, era a substituição das ligações por cabo de dispositivos fixos ou portáteis. Segundo o site oficial [Bluetooth SIG, 2011], o Bluetooth possui como características principais o baixo custo, o reduzido consumo energético e o baixo débito. Embora tenha sido imaginada como uma tecnologia para substituir os cabos, possui uma ampla gama de aplicações. Entre os dispositivos que utilizam Bluetooth podem-se incluir impressoras, *smartphones*, auriculares, relógios, entre outros.

3.7.1 Arquitetura

A arquitetura do Bluetooth v3.0 + HS (figura 3.8), consiste na combinação de um *host* com um ou mais *controllers* [Bluetooth SIG, 2009]. Um *host* encontra-se definido como todas as camadas em baixo dos perfis de utilização e em cima da interface Host Controller Interface (HCI). Por sua vez, um *controller* encontra-se definido como todas as camadas em baixo do HCI. Na especificação encontram-se definidos dois tipos de *controllers*:

- Basic Rate/Enhanced Data Rate (BR/EDR): que inclui as camadas Radio, *Baseband*, Link Manager e opcionalmente a camada HCI.
- Alternate MAC/PHYs (AMP): que inclui as camadas AMP PAL (Protocol Adaptation Layer), AMP MAC, Protocol Adaptation Layer (PAL) e opcionalmente a camada HCI.

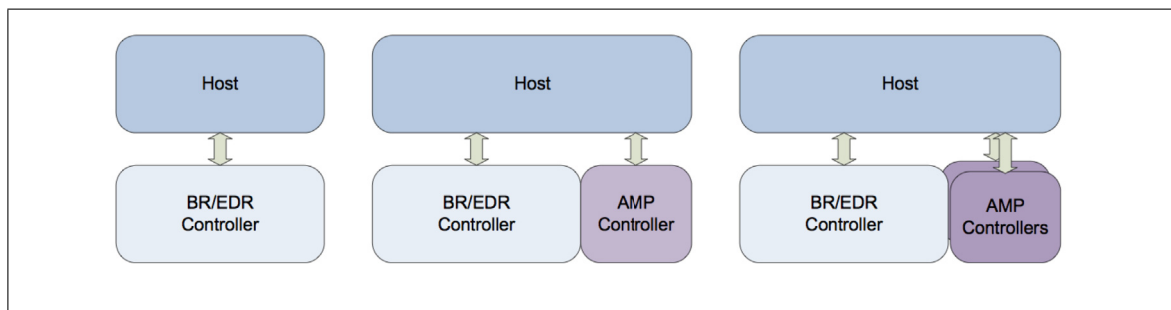


Figura 3.8: Arquiteturas possíveis do Bluetooth: BR/EDR, BR/EDR com um AMP e BR/EDR com vários AMPs [Bluetooth SIG, 2009].

3.7.2 Camada física

O Bluetooth funciona na banda ISM dos 2.4 GHz. Possui um total de 79 frequências com intervalos de 1 Mhz. De forma a minimizar as interferências, os dispositivos utilizam espalhamento espectral por saltos em frequência (FHSS). O Bluetooth utiliza modulação Gaussian Frequency Shift Keying (GFSK), e possui um débito de 1 Mbps no modo Basic Rate (BR) e 2 ou 3 Mbps no modo Enhanced Data Rate (EDR).

O alcance do Bluetooth depende da categoria do dispositivo. Tendo em consideração o alcance das ondas de rádio dos dispositivos Bluetooth, a especificação define as três classes de dispositivos representadas na tabela 3.1.

Tabela 3.1: Classes de dispositivos Bluetooth [Bluetooth SIG, 2011].

Classe	Potência Máxima	Alcance
Classe 1	100 mW (20 dBm)	até 100 metros
Classe 2	2.5 mW (4 dBm)	até 10 metros
Classe 3	1 mW (0 dBm)	\approx 1 metro

O canal físico encontra-se subdividido em unidades de tempo conhecidas como *slots*. Os dados transmitidos entre os dispositivos, são agrupados em pacotes posicionados nesses *slots*. A especificação disponibiliza a comunicação em ambos os sentidos, obtida através do aproveitamento da técnica Time Division Duplexing (TDD). Sendo que, salvo durante a fase de descoberta de dispositivos, a comunicação efetua-se unicamente entre mestre-escravo ou escravo-mestre [Bluetooth SIG, 2009].

3.7.3 Funcionamento básico

Durante o funcionamento típico é partilhado um canal de rádio com um grupo de dispositivos que se encontram sincronizados com um relógio comum. Um dos dispositivos que funciona como mestre disponibiliza a informação relativa à sequência de saltos. Os restantes dispositivos funcionam como escravos ajustando os seus relógios de acordo com a informação disponibilizada pelo mestre. Este é o modo fundamental de comunicação através de Bluetooth e denomina-se de *piconet*.

Uma *piconet* restringe a sua composição a um único mestre e a sete escravos ativos¹⁰. Várias *piconets* independentes podem coexistir no mesmo local sem interferência significativa. Cada *piconet* possui um dispositivo mestre independente que, por sua vez, disponibiliza uma sequência de saltos diferente.

Um dispositivo pode pertencer simultaneamente a mais do que uma *piconet*. No entanto, apesar de poder ser escravo em várias *piconets*, não pode ser mestre em mais do que uma delas. Quando um dispositivo pertence a duas ou mais *piconets* forma uma rede conhecida como *scatternet*. A especificação do Bluetooth define o conceito de *scatternet* de forma a interligar várias *piconets*, no entanto, não especifica nenhum protocolo de encaminhamento nem de comunicação de dados.

Na figura 3.9 apresentam-se as várias formas de comunicação disponibilizadas pela especificação.

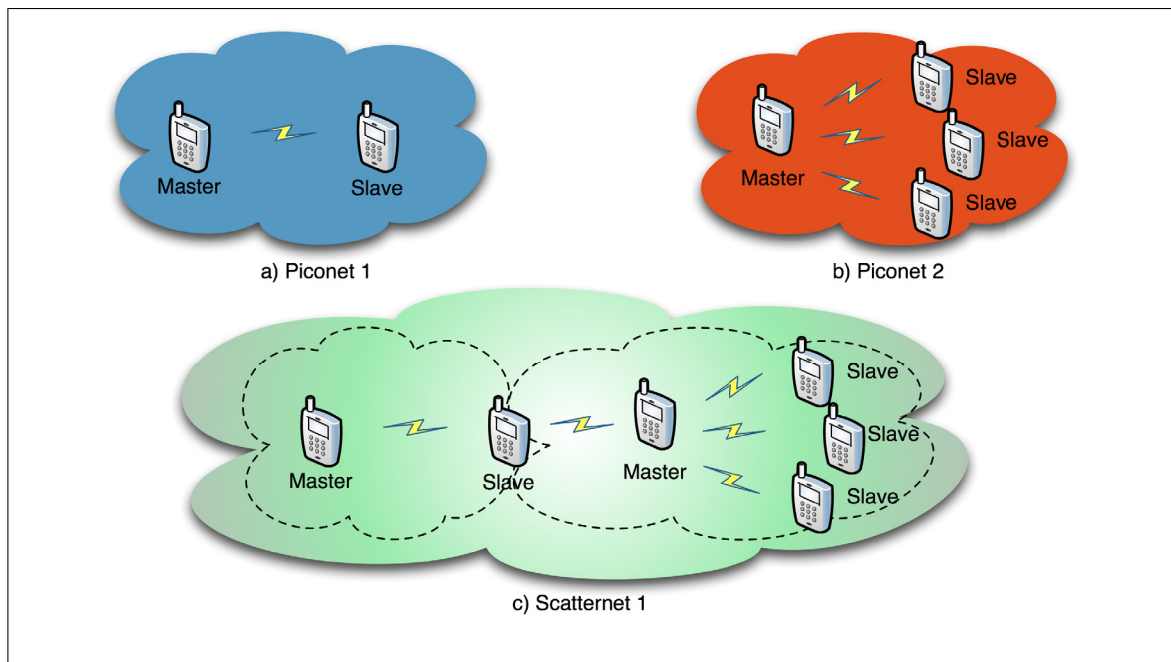


Figura 3.9: a) *Piconet* com apenas um dispositivo escravo; b) *Piconet* com três dispositivos escravos e c) Uma *scatternet* formada por três *piconets* [Bluetooth SIG, 2009].

¹⁰ Adicionalmente vários dispositivos podem permanecer ligados ao mestre em modo inativo (*parked*). Esses escravos em modo *parked* não se encontram ativos mas permanecem sincronizados com o mestre e podem-se tornar ativos sem passar novamente pelo processo de estabelecimento de ligação.

3.7.4 Formato dos pacotes

O formato genérico das mensagens enviadas pelos dispositivos Bluetooth encontra-se representado na figura 3.10. Cada pacote é constituído por três entidades: código de acesso, cabeçalho e *payload*. O código de acesso contém a identificação da *piconet* e possui 68 ou 72 bits. O cabeçalho possui 54 bits e inclui o endereço de um dispositivo escravo ativo na rede Bluetooth. O campo *payload* é utilizado para a informação realmente útil e pode conter até 2745 bits.

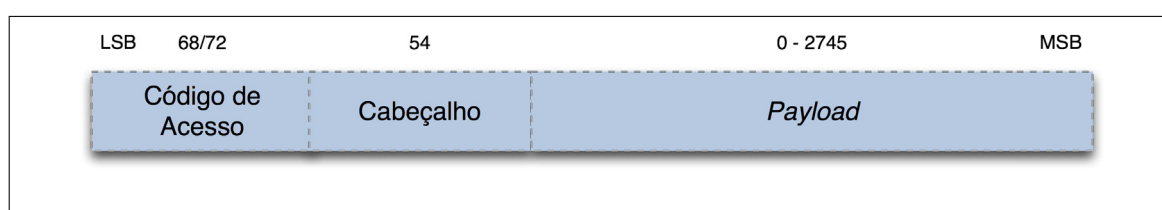


Figura 3.10: Formato genérico dos pacotes Bluetooth [Bluetooth SIG, 2009].

3.7.5 Estabelecimento da ligação

Para estabelecer uma ligação Bluetooth é necessário que os intervenientes executem duas fases [Thamrin & Sahib, 2009]: uma primeira fase denominada de *inquiry*, onde um dispositivo mestre procura por dispositivos ao seu alcance, e uma segunda fase conhecida como *page*, que permite a criação de um canal físico entre os dois dispositivos. Durante a primeira fase, é trocada a informação sobre os relógios para a sincronização dos dispositivos. Essa informação é utilizada na segunda fase para estabelecer a ligação entre os dispositivos.

Observando a figura 3.11 identificam-se as várias fases do estabelecimento de ligação entre dois dispositivos Bluetooth. Os dispositivos devem entrar no estado *inquiry* ou *inquiry scan* de forma a encontrem dispositivos ou serem encontrados por outros dispositivos respetivamente. Se um dispositivo que se encontra na fase de *inquiry scan* recebe um pacote IAC, realiza um período de *backoff* aleatório. No final do período de *backoff* responde com um pacote FHS que contém a sua identificação (BD_ADDR) e o valor do seu relógio.

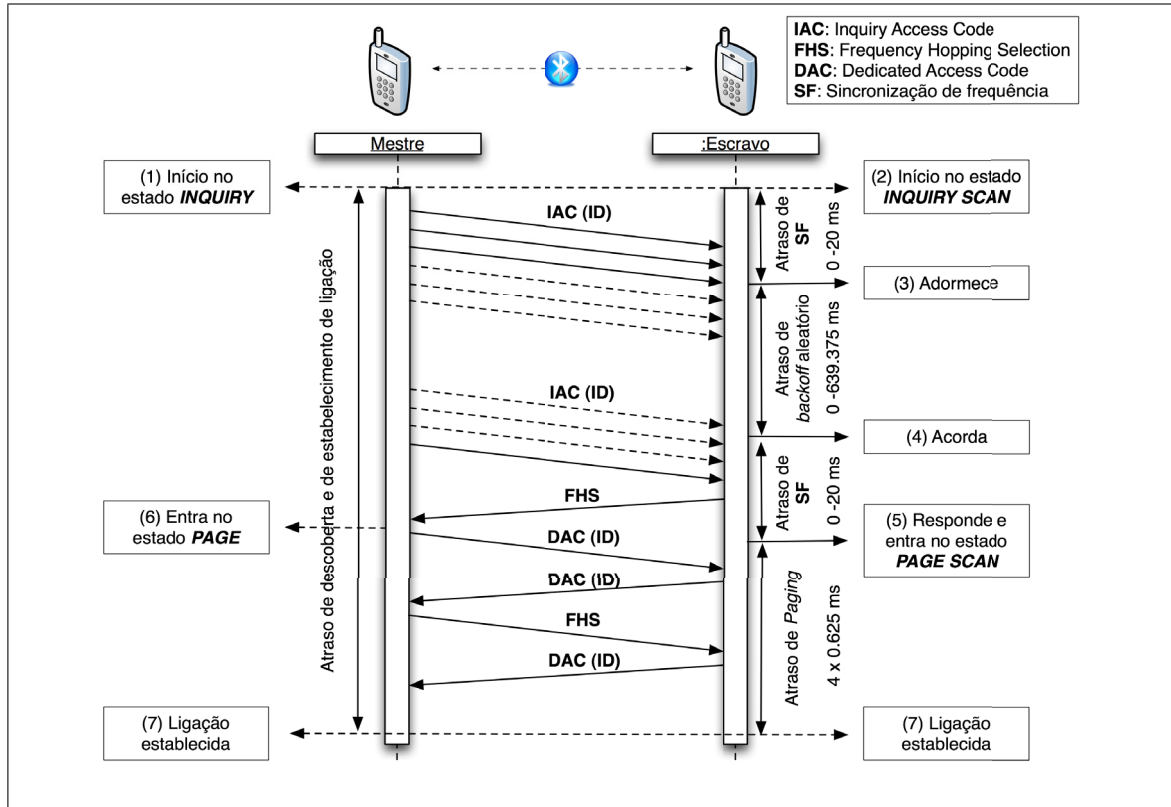


Figura 3.11: Estabelecimento de uma ligação Bluetooth [Thamrin & Sahib, 2009].

3.7.6 Bluetooth v4.0

Segundo o site oficial do Bluetooth [Bluetooth SIG, 2011], a versão 4.0 foi adotada em 30 de junho de 2010. O Bluetooth 4.0 é uma atualização à tecnologia de comunicação sem fios Bluetooth 3.0, lançada em 2009. A principal característica desta nova versão é o baixo consumo de energia. Esta capacidade possibilita a implementação do Bluetooth numa nova gama de dispositivos, permitindo que permaneçam em funcionamento durante vários meses ou anos com pequenas baterias. Os novos mercados para esses dispositivos incluem: saúde, desporto, segurança e entretenimento.

3.8 Conclusões

Várias especificações e tecnologias de redes sem fios surgiram nos últimos anos. Foram desenvolvidas tecnologias de comunicação tais como: IrDA, RFID, NFC, Zig-Bee, WiFi, Bluetooth. No entanto, todas essas tecnologias possuem limitações e restrições.

Em primeiro lugar, o RFID e o ZigBee não se encontram disponíveis nos dispositivos móveis que normalmente acompanham os utilizadores. O IrDA, tecnologia de comunicação presente em grande parte dos dispositivos móveis de alguns anos atrás, foi praticamente substituído pelo Bluetooth e já não se encontra disponível nos dispositivos móveis mais recentes. Por outro lado, através da análise realizada durante o presente capítulo, foi possível verificar algumas restrições na criação de redes Bluetooth, tais como o limite de ligações ativas numa *piconet* e o tempo necessário para estabelecimento de uma ligação.

Para o objetivo do projeto, pode-se concluir que tanto o Bluetooth como o WiFi, cumprem os requisitos, no sentido em que grande percentagem dos telefones móveis incorporam estas tecnologias. Embora outras tecnologias, que foram descritas anteriormente, possam ser utilizadas para a difusão de mensagens, devido às suas características técnicas, aos custos associados e à disponibilidade nos telefones móveis mais recentes, o Bluetooth e o WiFi são as tecnologias selecionadas para a realização do presente trabalho.

Capítulo 4

Estudo do Problema

A realização do trabalho proposto apresenta vários desafios: conseguir difundir um evento, em tempo útil, por um elevado número de telefones móveis, conseguir que o evento ocorra de forma sincronizada e lidar com a heterogeneidade dos dispositivos em termos de sistemas operativos, capacidades de processamento e de comunicação. Com o intuito de implementar o sistema, existem vários aspetos que devem ser considerados, assim como combinar soluções que resolvam os desafios existentes. Neste capítulo são apresentadas e discutidas diferentes abordagens para resolver os desafios do trabalho apresentado.

4.1 Abordagem sem infraestrutura

Recentemente, a grande penetração de novas tecnologias em dispositivos móveis, como Bluetooth e IEEE 802.11, possibilita a criação de redes do tipo Mobile Adhoc Networks (MANET) [Chlamtac et al., 2003]. Na abordagem que se representa na figura 4.1, os dispositivos podem, livre e dinamicamente, organizarem-se em topologias de rede arbitrárias e temporárias, possibilitando a comunicação entre eles sem a necessidade de uma infraestrutura previamente definida [Frodigh et al., 2000].

Neste caso em particular, os espectadores/utilizadores, apenas necessitam de se encontrar dentro da área de alcance para participarem no evento. Ao contrário de outras tecnologias de transmissão de informação (como por exemplo SMS/MMS), o Bluetooth permite dirigir a informação para todas as pessoas que se encontram na área em vez de um grupo de pessoas conhecidas. Além disso, o custo de transmissão de informação utilizando a tecnologia Bluetooth é nulo.

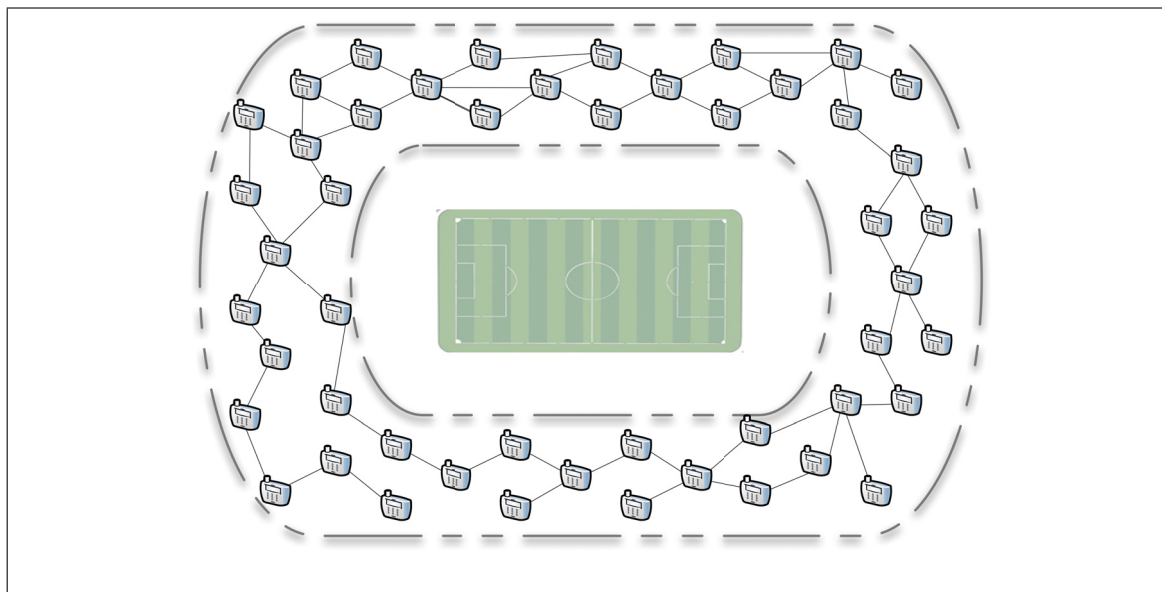


Figura 4.1: Abordagem sem infraestrutura.

A grande vantagem das redes *ad-hoc*, encontra-se na maior flexibilidade, mesmo quando não existe uma infraestrutura de comunicação ou esta não se encontra disponível. Por estas características, podem ser utilizadas em locais onde é necessária uma rápida instalação, como por exemplo eventos artísticos, locais atingidos por catástrofes naturais e operações militares [Chlamtac et al., 2003].

Uma das desvantagens da combinação da tecnologia Bluetooth com este modelo, é a grande complexidade dos nós que constituem a rede. Apesar do Bluetooth já possuir uma certa maturidade, o estabelecimento da ligação, mesmo nas versões mais recentes, consome ainda demasiado tempo [Thamrin & Sahib, 2009; Asthana & Kalofonos, 2005] e, mesmo depois de estabelecida a ligação, cada dispositivo tem a sua forma de lidar com as mensagens recebidas [Aiello et al., 2009].

Considerando os detalhes técnicos do Bluetooth (descrito no capítulo 3), deduz-se que não se encontra direcionado para o envio de informação para um grande grupo de utilizadores. Um dispositivo Bluetooth, apenas pode enviar dados para sete dispositivos em simultâneo. Para evitar este problema, a especificação do Bluetooth define o conceito de *scatternet* de forma a interligar várias *piconets*. Contudo, as restrições e as propriedades da criação de *scatternets*, apresentam vários desafios na formação eficiente de uma rede *ad-hoc*. Nos últimos tempos, surgiram vários algoritmos procurando otimizar o processo de criação de *scatternets*, no entanto, todos eles aumentam a complexidade nas aplicações dos dispositivos.

4.1.1 Estabelecimento da ligação

Embora o Bluetooth seja uma tecnologia bastante atraente para a resolução do problema proposto, possui uma restrição fundamental, que torna difícil a sua utilização neste contexto. O problema, é que o processo de descoberta de dispositivos e serviços Bluetooth consome demasiado tempo [Handurukande et al., 2006]. Cada dispositivo necessita de procurar constantemente por novos dispositivos ao seu alcance e, quando um dispositivo se encontra à procura de outros dispositivos ou serviços, não pode ser detetado por outros dispositivos. Assim, quanto mais tempo um dispositivo consumir à procura de outros dispositivos, mais tempo permanece invisível para outros. No pior cenário, se a procura for realizada de forma simultânea, os dispositivos podem nunca se encontrar [Handurukande et al., 2006].

Uma eventual alternativa à procura simultânea é esperar um período de tempo aleatório antes da procura [Handurukande et al., 2006]. No entanto, como neste caso é esperada uma elevada densidade de dispositivos, o tempo necessário para o estabelecimento da ligação torna-se demasiado longo e insuportável.

4.1.2 Poluição em redes Bluetooth

O problema em redes Bluetooth conhecido como “poluição em redes Bluetooth”, é causado pela presença de um grande número de dispositivos desconhecidos¹, na mesma área de cobertura. Segundo Asthana & Kalofonos [2005], uma elevada densidade de dispositivos na mesma área de cobertura afeta o desempenho do processo de estabelecimento de ligação entre dispositivos Bluetooth. Contudo, este problema não se refere aos problemas causados por interferências rádio (como por exemplo colisões e retransmissões). Este problema é provocado pela interação negativa durante as operações básicas de pesquisa e de ligação entre os dispositivos. Estes dispositivos desconhecidos realizam pedidos aos dispositivos da aplicação, levando-os a ficar temporariamente ocupados, uma vez que se encontram a responder a outros pedidos [Asthana & Kalofonos, 2005].

¹ O conceito de dispositivo desconhecido, refere-se a dispositivos que não cooperam, ou seja, dispositivos que não se encontram no contexto da mesma aplicação.

4.1.3 Criação de *scatternet*

Uma *scatternet* é formada através da ligação de duas ou mais *piconets*². Um dispositivo apenas pode exercer o papel de mestre numa *piconet*, no entanto, é permitida a sua participação em várias, como escravo. De forma a interligar várias *piconets*, a especificação do Bluetooth define o conceito de *scatternet*, porém, não especifica nenhum protocolo de ligação nem de encaminhamento. Têm surgido várias abordagens para resolver o problema da formação de *scatternets*. No entanto, muitas dessas abordagens dependem de entidades centralizadas e, portanto, são de utilização limitada.

A primeira questão é como construir ligações entre os dispositivos. Isto significa que é necessário determinar os papéis dos intervenientes, ou seja, têm de ser escolhidos os dispositivos que vão atuar como mestre e como escravo e, além disso, têm de ser selecionados os dispositivos que vão realizar a interligação entre as *piconets*.

A especificação do Bluetooth define que um escravo apenas pode transmitir para o seu mestre. Além disso, uma vez que os nós podem ser móveis, as condições do canal de transmissão podem variar constantemente. De forma a permitir uma comunicação eficiente, devem ser utilizados protocolos de encaminhamento. Diferentes métodos podem ser utilizados para o encaminhamento das mensagens. Segundo Haas et al. [2002], protocolos como LAR, GPSR e DREAM, assumem que os dispositivos possuem GPS, de forma a obter a sua localização, outros, como DSR, AODV, ZRP e TORA, não fazem essa suposição. Geralmente, esses protocolos são baseados em *flooding*³ (com algumas otimizações). No entanto, apesar de várias otimizações, muitas mensagens de encaminhamento são propagadas desnecessariamente.

Mesmo depois de todo o tempo necessário para a formação das *scatternets*, é ainda necessária uma camada superior, que implemente um protocolo de encaminhamento complexo e redundante.

² Uma vez que cada *piconet* possui uma sequência de salto em frequência única, a interligação de *piconets* é possível quando um dispositivo pertence a duas ou mais *piconets*.

³ O algoritmo básico de *flooding* inicia-se quando um dispositivo difunde uma mensagem para todos os seus vizinhos. Cada um desses vizinhos, retransmite a mensagem para todos os seus vizinhos. O algoritmo continua até que todos os dispositivos da rede tenham recebido a mensagem [Williams & Camp, 2002].

4.1.4 Partição da rede

A partição da rede refere-se à divisão da rede em dois ou mais grupos de dispositivos que não podem comunicar uns com os outros. Sendo a mobilidade uma das principais características de uma MANET, a distribuição dos dispositivos muda frequentemente. Deste modo, a topologia de rede encontra-se em constantes alterações e os dispositivos podem-se juntar ou separar. Portanto, em regiões com elevada densidade de dispositivos, existe maior probabilidade de ocorrência de colisões, e em regiões de baixa densidade, pode ocorrer a separação dos dispositivos, tornando a comunicação difícil ou mesmo impossível [Khelil et al., 2005].

Por outro lado, a mobilidade associada a estas redes permite que os dispositivos móveis armazenem e transportem as mensagens (*store-and-forward*). Os dispositivos armazenam as mensagens e retransmitem sempre que o destino se torne, mais facilmente, acessível. Com este mecanismo é possível minimizar o problema da partição da rede. No entanto, de acordo com Khelil [2007], as aplicações que utilizam este mecanismo devem ser tolerantes a atrasos. Por conseguinte, uma vez que não existe uma elevada mobilidade associada aos dispositivos alvo, este mecanismo não se encontra aplicável no cenário pretendido.

4.1.5 Diversidade de dispositivos

Os dispositivos utilizados podem ser de uma grande variedade de marcas e modelos, cada um com a sua própria maneira de lidar com os conteúdos recebidos através do Bluetooth. Desta forma, não existe garantia que cada cliente recebe o conteúdo como pretendido. Alguns *smartphones* exigem a configuração de um canal seguro, através de um Personal Identification Number (PIN), antes de iniciar a transmissão de informação. Mesmo existindo algumas soluções para ultrapassar este mecanismo de segurança (*BlueSnarfing*, *Bluejacking*, etc.), todas elas, de uma forma ou de outra, violam a privacidade do utilizador.

Deixar os dispositivos com o Bluetooth visível pode ser um risco de segurança. Para evitar que alguns utilizadores se sintam invadidos, pode-se optar por uma transmissão passiva. Neste modelo, os utilizadores tomam a iniciativa de pedir a informação que pretendem. No entanto, esta solução exige a existência de uma entidade, conhecida por todos os elementos da rede, que seja responsável pela receção (e respetiva resposta) dos pedidos dos diversos dispositivos.

4.1.6 Sincronização

Como em todos os sistemas distribuídos, a sincronização dos relógios é uma componente importante. Existem várias razões que levam à necessidade de sincronização temporal entre os intervenientes de um sistema. Neste caso em particular, os dispositivos necessitam de sincronizar as suas operações de forma a ser possível criar um determinado evento coordenado entre todos eles.

São necessários *timestamps*⁴ para ordenar as operações de um modo coerente com o mundo real. No entanto, o problema da imprecisão de sincronização dos relógios físicos dos dispositivos, leva a uma falta de precisão dos *timestamps*. Os principais fatores de falta de sincronização dos relógios são: diferentes relógios; diferentes frequências dos cristais e variação no atraso das mensagens [Scholz et al., 2007].

Uma vez que nem todos os dispositivos contêm acesso à Internet, a obtenção externa de relógio torna-se inviável. Os tradicionais mecanismos de sincronização, como Network Time Protocol (NTP) e GPS não são adequados para este projeto devido à sua complexidade, ao consumo energético, ao custo e à disponibilidade [Lasassmeh & Conrad, 2010]. O mecanismo NTP é adequado para a sincronização de computadores que possuam ligação à Internet, contudo, não foi desenvolvido tendo em conta as limitações dos dispositivos móveis. Além disso, no contexto da abordagem sem infraestrutura, os dispositivos não se encontram constantemente ligados à Internet. Em relação ao serviço de GPS, este pode não se encontrar disponível em todo o lado (por exemplo no interior de edifícios).

A solução para este problema, poderia passar pela difusão de uma mensagem pela rede, a notificar que deve ocorrer um evento passado x tempo (ou a determinadas horas) e dar tempo para que essa mensagem seja difundida por todos os dispositivos. Apesar disso, sem ter em conta os problemas apresentados anteriormente (estabelecimento da ligação, criação de *scatternets* e partição da rede), esta solução aumenta a complexidade nos dispositivos e além disso, a sincronização de todos os dispositivos pode consumir demasiado tempo.

⁴ *Timestamps* são marcas temporais utilizadas para armazenar a hora e a data de um determinado acontecimento.

4.2 Abordagem com infraestrutura

O SyncEvent, baseado de redes WiFi, explora a infraestrutura de um local através de um dispositivo móvel. Como ilustrado na figura 4.2, os dispositivos móveis aproveitam a infraestrutura de redes WiFi de um determinado local para comunicar com o servidor.

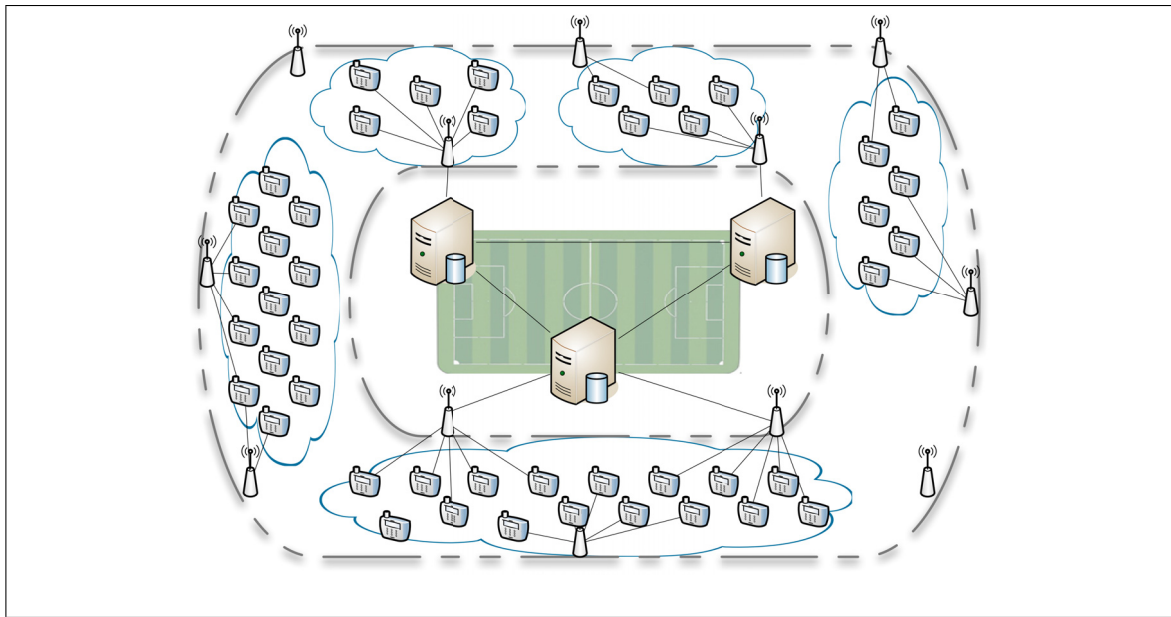


Figura 4.2: Abordagem com infraestrutura.

A solução baseada em infraestrutura apresenta-se como uma solução comercialmente mais apelativa. Esta abordagem, possibilita um controlo na realização do evento e, além disso, existe um sistema central que permite evitar a ativação de vários eventos em simultâneo.

Com a localização dos dispositivos no estádio é possível criar efeitos “avançados”. A solução baseada em infraestrutura permite obter a localização dos dispositivos e, ativando apenas os telefones que se encontrem em determinadas posições do estádio, é possível “desenhar figuras”, possibilitando a utilização do sistema como um meio de publicidade.

A abordagem baseada em infraestrutura permite impulsionar a adoção do sistema. Para incentivar as pessoas a utilizar o sistema podem ser realizados sorteios de recompensas entre os participantes.

4.2.1 Operação básica

Quando ligado, um dispositivo WiFi procura por canais disponíveis, para encontrar redes ativas onde se encontrem pacotes *beacons* a serem transmitidos. Em seguida, seleciona uma rede, e caso seja uma rede baseada em infraestrutura autentica-se com o ponto de acesso, associando-se a ele em seguida. No caso de se encontrar implementado um mecanismo de segurança, será necessário uma fase adicional de autenticação antes do dispositivo poder participar na rede [Ferro & Potorti, 2005].

4.2.2 Complexidade e custo

Uma das desvantagens das redes *ad-hoc*, principalmente quando a comunicação é realizada através de intermediários, é a grande complexidade dos nós que constituem a rede. Nesse tipo de redes, onde não existe uma infraestrutura que controle o meio de difusão, cada nó deve possuir mecanismos de controlo de acesso ao meio, mecanismos que lidem com o problema dos terminais expostos ou escondidos e ainda mecanismos que forneçam alguma qualidade de serviço. A complexidade do encaminhamento aumenta numa rede *ad-hoc*, pois, para além do acesso ao meio ser decidido por todos os nós participantes na rede, cada um desses nós deve exercer a função de encaminhador.

A abordagem com infraestrutura diminui a complexidade nas aplicações dos dispositivos. Tipicamente, as redes sem fios baseadas em infraestrutura são parte de uma rede mais extensa, cujo núcleo é normalmente uma rede cablada. Estas redes encontram-se configuradas de forma a que os dispositivos não possam comunicar diretamente entre si, realizando a comunicação apenas com os pontos de acesso, que servem de interface entre o segmento sem fios e o resto da rede. Os pontos de acesso não só proporcionam o acesso a outras redes como também possuem funções de encaminhamento, controlo de acesso ao meio, entre outros. Como o ponto de acesso é o único responsável pelo controlo de acesso ao meio, para além de possuir autoridade para regular o acesso por parte dos dispositivos, garante também a comunicação para todos eles. Assim, um dispositivo que deseje transmitir deve aguardar a autorização do ponto de acesso.

Uma das grandes vantagens do modelo baseado em infraestrutura é a existência de um sistema central que, para além de evitar a ativação de vários eventos em simultâneo, impede ainda a criação de várias redes no mesmo local. A adoção desta

abordagem para a resolução do problema implica a criação de uma infraestrutura. Contudo, este facto não é muito limitador dado que, hoje em dia, os seus custos são reduzidos.

4.2.3 Localização dos dispositivos

Atualmente existem inúmeras soluções de localização, quer para ambientes interiores como exteriores. No entanto, todas possuem as suas próprias características e limitações. O sistema GPS permite uma localização com um grau de precisão aceitável. No entanto, segundo Xiao et al. [2011], este tipo de posicionamento exige a existência de linha de vista entre o recetor e os satélites GPS. Para além disso, em ambientes interiores o sinal de GPS é gravemente atenuado. Assim, o sistema encontra-se dependente da localização do recetor e da disponibilidade dos satélites.

Por outro lado, tem-se assistido ao aparecimento de técnicas de localização direcionadas a ambientes interiores baseadas em tecnologias como WiFi, Bluetooth, RFID, etc. É com facilidade que se encontram soluções de localização baseadas em técnicas como: Received Signal Strength Indication (RSSI), Time of Arrival (TOA), Time Difference of Arrival (TDOA) e Angle of Arrival (AOA). No entanto, segundo Xiao et al. [2011], são necessários mais esforços, no sentido de melhorar a precisão de localização, reduzir os custos dos equipamentos e diminuir o consumo energético.

O elevado custo computacional, associado às várias técnicas de posicionamento existentes nos dias de hoje, a complexidade de implementação e a dependência do contexto, faz com que a sua utilização no âmbito deste projeto não seja a solução mais adequada. Em vez disso, propõe-se uma abordagem onde os dispositivos possam ser localizados de forma explícita ou estimada. De forma explícita, cabe ao utilizador indicar a sua posição no estádio. Em alternativa, considera-se a utilização do Bluetooth para descobrir os seus vizinhos e, fundamentado nessa informação, ser capaz de estimar a sua localização.

4.2.4 Difusão dos eventos

A distribuição das mensagens associadas a um determinado evento e a um local específico (físico), onde o evento se encontra ativo, deve ocorrer de tal forma que seja possível alcançar, em tempo útil, a totalidade (ou pelo menos a grande maioria) dos interessados em participar no evento. A transmissão da informação ocorre dentro

de um espaço limitado, após a solicitação dos dispositivos móveis que se encontram dentro da área de cobertura. Os pedidos podem ser efetuados através de diversos suportes de transmissão, e recebidas exclusivamente por um dispositivo móvel na posse de um utilizador que se encontre no local.

4.3 Abordagem alternativa

Uma possibilidade alternativa para a resolução do problema é, tal como na abordagem anterior, tirar partido não só de uma infraestrutura WiFi presente no local mas também do Bluetooth. A maior diferença é que esta é uma solução que não considera a localização (explícita) dos dispositivos no local do evento, sendo o posicionamento dos dispositivos efetuado com base na proximidade que possuem com outros dispositivos.

Utilizando o Bluetooth pode obter-se informação dos dispositivos existentes nas proximidades e fornecer essa informação ao servidor. Um dispositivo com Bluetooth pode obter a identificação de outros dispositivos à sua volta e, em caso de sucesso, informa o servidor. Assim, o servidor deve ser capaz de posicionar os dispositivos, processando os dados recebidos e criando um grafo⁵ através da relação de proximidade entre os dispositivos.

Este sistema permite que o servidor produza um grafo em função da proximidade que o utilizador tem com outros dispositivos. Por exemplo, um grupo de dispositivos que se encontra num determinado local pode obter a identificação dos seus vizinhos onde o resultado poderia ser: o dispositivo d_a está próximo de d_b , d_c e d_d . Por sua vez, o dispositivo d_b encontra-se próximo de d_e e o dispositivo d_c consegue obter a identificação dos dispositivos d_e e d_f , etc. Torna-se assim praticável a construção do grafo que se ilustra na figura 4.3, o qual é baseado na vizinhança identificada entre os diversos dispositivos presentes num local.

4.3.1 Vantagens

O Bluetooth é apenas utilizado para detetar outros dispositivos à sua volta. A adoção desta abordagem permite, por exemplo, eliminar por completo a necessidade

⁵ Grafo é uma representação gráfica constituída por pontos e segmentos de reta entre eles. Neste caso em particular os pontos do grafo e os segmentos de reta correspondem, respetivamente, aos dispositivos móveis e à existência de uma relação de proximidade entre eles.

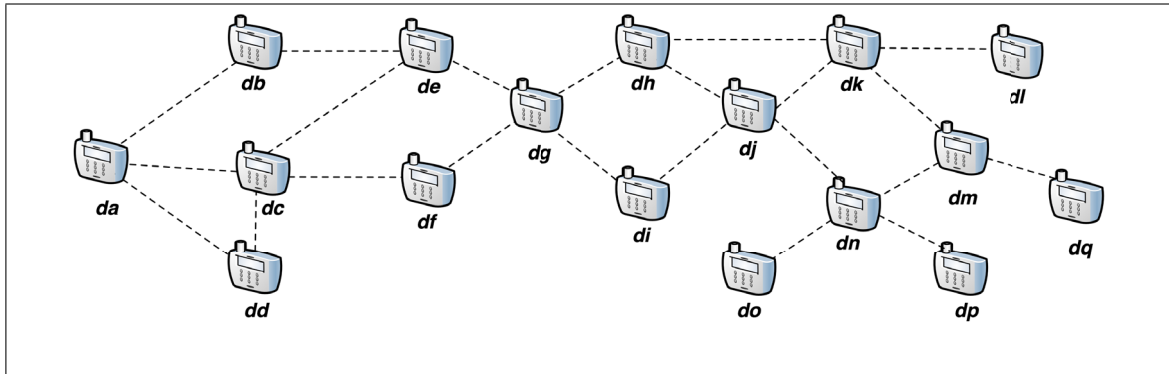


Figura 4.3: Grafo de proximidade entre dispositivos móveis.

de estabelecer ligações Bluetooth entre os dispositivos móveis. Estes apenas necessitam de descobrir os dispositivos que se encontram no alcance rádio do Bluetooth, sendo transmitidas ao servidor as identificações obtidas (ou seja, os endereços MAC das interfaces de Bluetooth). Desta forma, em vez de serem os dispositivos a comunicar a localização para outros dispositivos, passa-se a carga computacional para o servidor que, através do conhecimento transmitido por cada dispositivo, vai obtendo conhecimento da proximidade entre todos eles, produzindo um grafo.

Além de se extinguir a necessidade do estabelecimento de ligações Bluetooth, os utilizadores deixam de ser perturbados com constantes pedidos de associação, para que disponibilizem as suas informações de localização a outros dispositivos.

4.3.2 Limitações

Apesar das vantagens verificadas com esta abordagem, identifica-se um conjunto de limitações:

- **Partição da rede:** Uma vez que os espectadores podem ocupar os lugares sem restrições, surge a possibilidade da formação de diversos grupos de dispositivos que, devido ao curto alcance do Bluetooth, não podem comunicar uns com os outros.
- **Aumento da carga computacional do servidor:** Visto que se considera comunicar ao servidor apenas a informação dos dispositivos que se encontram na proximidade, o servidor terá que lidar com estruturas de dados complexas, que relacionem os dispositivos aos seus vizinhos.

- **Perda da noção de localização no espaço:** Unicamente com o Bluetooth não é possível conhecer as posições relativas⁶ entre os dispositivos. Deste modo, o servidor apenas possui a informação da relação de proximidade entre os dispositivos. No entanto, a localização no espaço é imprescindível para a configuração de um evento, uma vez que não é possível difundir o evento de forma coordenada sem ter noção da disposição dos dispositivos.

A abordagem alternativa, apresentada durante esta secção, não contempla a difusão dos eventos. Esta abordagem considera apenas o processo de localização dos dispositivos. Assim, para a difusão dos eventos pelos dispositivos móveis, teria de se utilizar o WiFi como descrito na secção 4.2.

4.4 Conclusões

Mediante o cenário apresentado, os alvos do presente são dispositivos móveis, mais precisamente *smartphones*. Como a grande maioria desses dispositivos possui a tecnologia Bluetooth, foi equacionada a sua utilização no âmbito deste trabalho. No entanto, do ponto de vista tecnológico, facilmente se apercebe que a tecnologia Bluetooth, mesmo após grandes progressos, pode ainda ser considerada subaproveitada, uma vez que possui características que dificultam a criação de redes móveis *ad-hoc* em grande escala.

Todas as soluções apresentadas tem vantagens e desvantagens. Contudo, para esta situação em particular, as restrições apresentadas pela abordagem sem infraestrutura fazem com que essa solução não seja a mais adequada para esta situação. Assim, optou-se por uma abordagem baseada em infraestrutura, que combina a disponibilidade da tecnologia Bluetooth com a maturidade da WiFi.

A abordagem com infraestrutura cumpre os requisitos necessários e torna-se mais apelativa do ponto de vista comercial. Através da localização dos dispositivos no estádio, podem-se produzir efeitos “avançados”, possibilitando a utilização do sistema como um meio de publicidade. Para além disso, apesar do Bluetooth possuir algumas restrições na criação de redes em grande escala, pode ser utilizado para detetar outros dispositivos à sua volta. Assim, a informação de localização disponibilizada pelos vizinhos, pode ser utilizada para determinar, aproximadamente, a localização dos dispositivos.

⁶ As principais posições relativas são: “em cima”, “em baixo”, “à direita” e “à esquerda”.

Capítulo 5

Especificação do Sistema

Neste capítulo é realizada uma descrição da arquitetura do sistema de eventos sincronizados em sistemas móveis fisicamente próximos, designado de SyncEvent. O capítulo começa por especificar a arquitetura global do sistema proposto. Esta arquitetura, encontra-se preparada para suportar a ocorrência de vários eventos em diferentes locais. De seguida, apresenta-se uma instanciação simplificada da arquitetura, direcionada apenas para a criação de eventos num único local. Neste capítulo é ainda realizada uma descrição do protocolo de comunicação que, especificando a estrutura das mensagens, torna possível o entendimento entre os diferentes intervenientes do sistema.

5.1 Arquitetura global

O SyncEvent utiliza um servidor central para efetuar o controlo de todo o processo de geração de eventos. Para participarem num determinado evento os utilizadores podem comunicar com o servidor através de uma infraestrutura WiFi presente no local ou, em alternativa, diretamente através de uma outra tecnologia de acesso à Internet presente nos dispositivos móveis dos utilizadores, como por exemplo UMTS ou High Speed Packet Access (HSPA).

Salienta-se que um dos objetivos desta dissertação é a criação de um sistema escalável, capaz de difundir a informação de um determinado evento para um grande grupo de dispositivos móveis, através de uma tecnologia de comunicação sem fios. Para atingir esse objetivo, é necessário:

- Um servidor com ligação à Internet ou a uma infraestrutura WiFi;

- Uma base de dados para guardar a informação necessária;
- Um grupo de utilizadores que contenha a aplicação SyncEvent instalada nos seus dispositivos móveis, e que estes possuam uma tecnologia de comunicação sem fios¹.

Para a arquitetura global assume-se que será necessário mais do que um ponto de acesso para cobrir toda a área de cada local e, eventualmente, mais do que um servidor para responder da melhor forma às necessidades de todos utilizadores.

A figura 5.1 ilustra como os diferentes componentes se relacionam. Os servidores guardam a informação dos vários eventos e dos vários locais. Os dispositivos móveis, por sua vez, são responsáveis por estabelecer uma ligação com o servidor, através da qual devem informar a sua localização no local para que, posteriormente, possam obter informações sobre os eventos disponíveis.

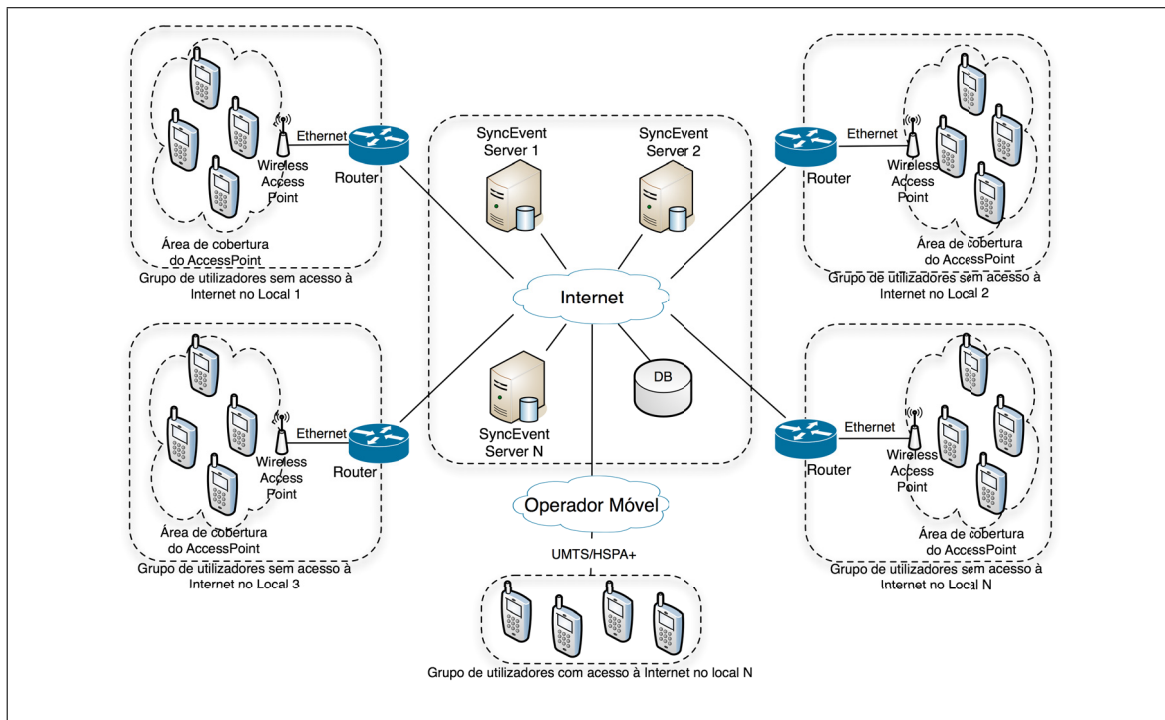


Figura 5.1: Arquitetura global.

Os dispositivos podem estabelecer ligações com outros dispositivos ao seu alcance. Uma vez que se considera a localização dos vizinhos para estimar localização de um

¹ O acesso à aplicação pode ser realizado via WiFi, UMTS/HSPA+ ou através de outra tecnologia de comunicação sem fios disponível nos dispositivos móveis dos utilizadores.

dispositivo, estes podem estabelecer ligações entre eles com o objetivo de se obter informações relativas à sua localização. As informações de localização dos vizinhos são enviadas ao servidor que, baseado nessa informação, calcula aproximadamente a localização do dispositivo.

5.2 Arquitetura do projeto

A arquitetura descrita na secção anterior é bastante abrangente. Um sistema multiservidor, com replicação da informação de forma a prestar um melhor serviço aos utilizadores, não se encontra no âmbito desta dissertação. Para este projeto é apenas considerado um local onde existe uma infraestrutura sem fios, que permite aos visitantes, com dispositivos móveis que suportem WiFi e que contenham a aplicação SyncEvent instalada, a possibilidade de participarem num determinado evento.

A figura 5.2 representa a relação entre os elementos que constituem o SyncEvent e que permitem oferecer, aos visitantes de um local, a possibilidade de participarem em eventos com os seus dispositivos móveis.

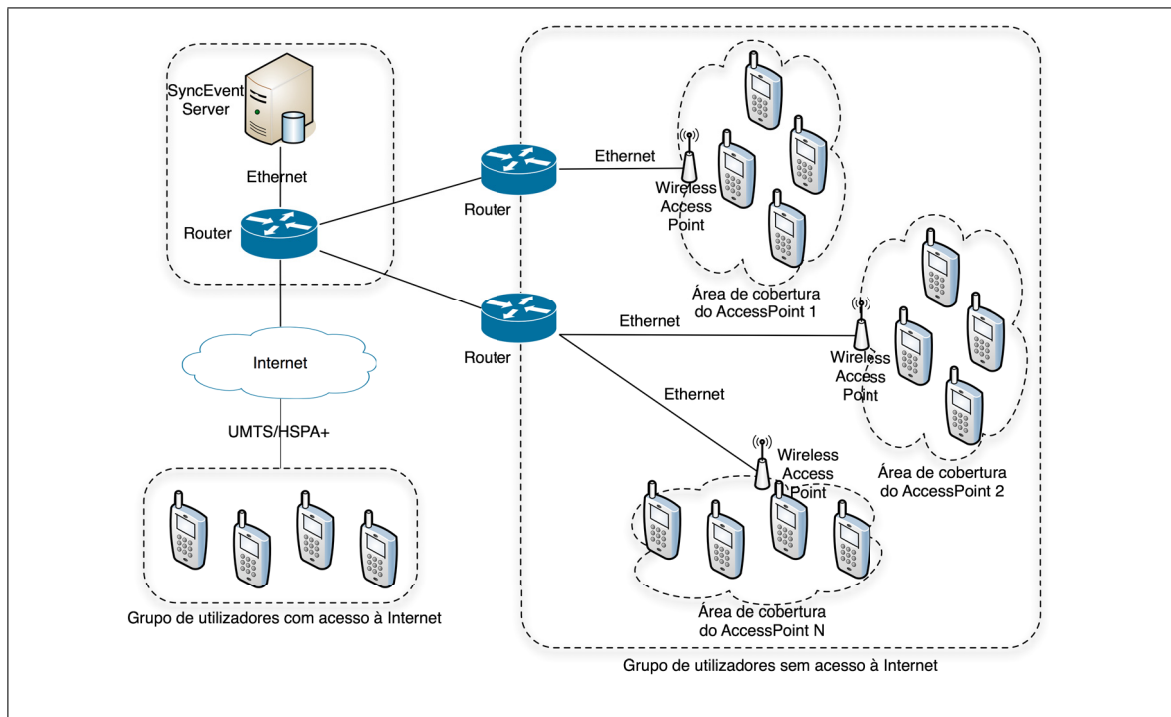


Figura 5.2: Arquitetura do projeto.

5.3 Camadas do sistema

O sistema consiste em dois subsistemas conforme ilustrado na figura 5.3. Cada subsistema encontra-se dividido em camadas que por sua vez se encontram divididas em módulos.

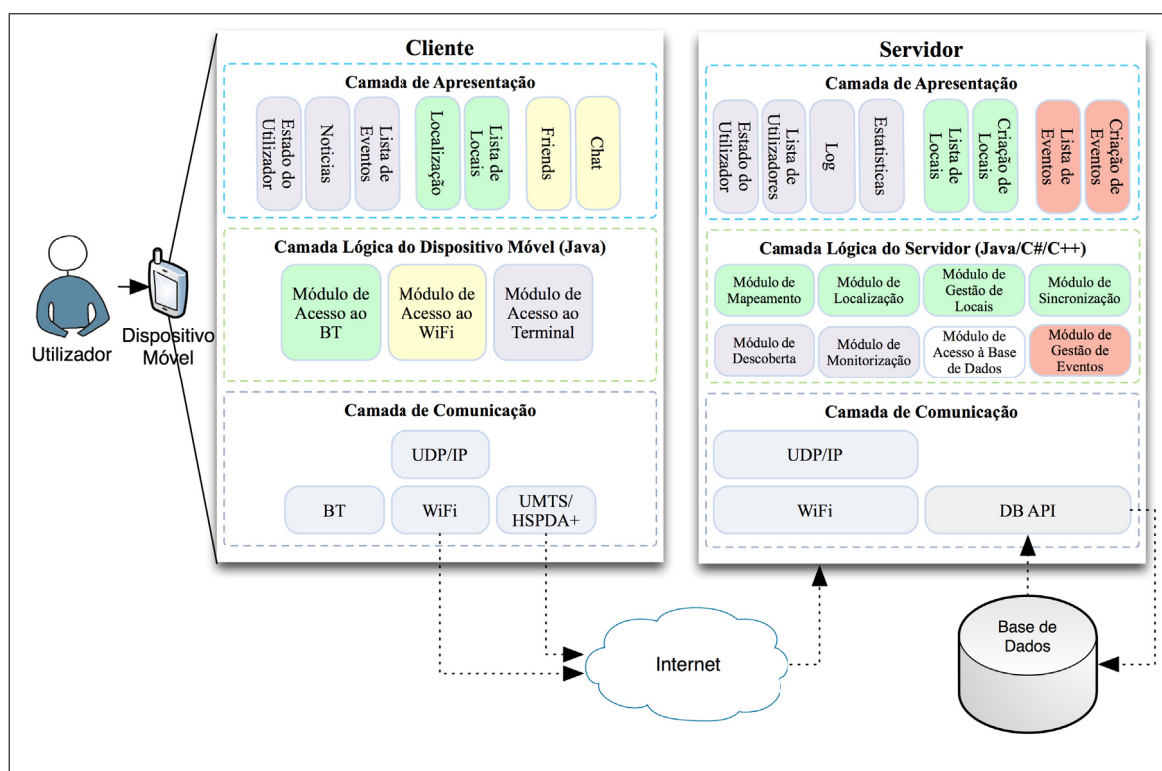


Figura 5.3: Camadas do sistema.

Do lado do subsistema do dispositivo móvel, a camada de apresentação é responsável pela apresentação das informações importantes, incluindo informações de localização e de eventos.

A camada lógica do dispositivo móvel é composta por três módulos: módulo de acesso ao Bluetooth; módulo de acesso ao WiFi e módulo de acesso ao terminal. O módulo de acesso ao Bluetooth deve conseguir detetar e estabelecer ligações com dispositivos ao seu alcance. As ligações estabelecidas com sucesso são utilizadas para obter informações de localização. Essas informações são enviadas ao servidor para que, baseado nessa informação, estime a localização do dispositivo.

O módulo de acesso ao terminal, contém as funcionalidades necessárias para a realização dos eventos, como por exemplo o acesso à iluminação do visor/teclado dos

dispositivos móveis, para produzir o efeito visual, e acesso ao vibrador, para que seja possível alertar o utilizador da aproximação de um evento.

Na camada lógica do servidor, os módulos de localização e de descoberta são utilizados em conjunto, para determinar a posição dos diversos dispositivos. O módulo de localização trata a informação de localização enviada por um utilizador que conhece o seu lugar e pretende ser localizado de forma precisa. Por outro lado, o módulo de descoberta é responsável pelo tratamento da informação enviada por um utilizador, que pretende ser localizado de forma estimada.

Os módulos de mapeamento e de sincronização encontram-se respetivamente responsáveis pelo processamento da informação referente aos locais e pela sincronização dos dispositivos. A seguir descrevem-se os módulos mais importantes do sistema.

5.3.1 Módulo de mapeamento

O módulo de mapeamento efetua o processamento da informação referente aos locais. Os locais para a realização desses eventos, são locais onde normalmente se concentra, em simultâneo, uma grande quantidade de pessoas (como por exemplo auditórios, estádios de futebol, concertos, entre outros).

Os locais destinados à ocorrência dos eventos possuem diversas configurações e diversas características. No entanto, todos eles, de uma forma ou de outra, possuem os seus lugares identificados. Por exemplo, num estádio de futebol, como se ilustra na figura 5.4, os lugares destinados aos espectadores encontram-se identificados através da bancada, do andar, da secção, da fila e do lugar.

As configurações de cada local podem-se encontrar num documento de texto. No caso do exemplo do estádio representado na figura 5.4, existem quatro bancadas, todas elas com três andares. Cada andar possui secções que por sua vez possuem filas e estas os lugares. Ou seja, cada lugar é identificado através da bancada, do andar, da secção, da fila e do lugar. Desta forma, o documento de configurações de cada bancada contém a identificação dos respetivos andares e das respetivas secções. Por sua vez, cada secção contém as suas filas e cada fila os respetivos lugares.

As configurações dos locais são diferentes de local para local. Em certos locais (como é o caso do estádio), os pisos superiores possuem mais lugares do que os pisos inferiores. De forma a alinhar os lugares dos espectadores, é necessário “compensar” os lugares a menos dos pisos inferiores. No documento de configuração, o equilíbrio dos lugares é realizado através da utilização da letra “X”.

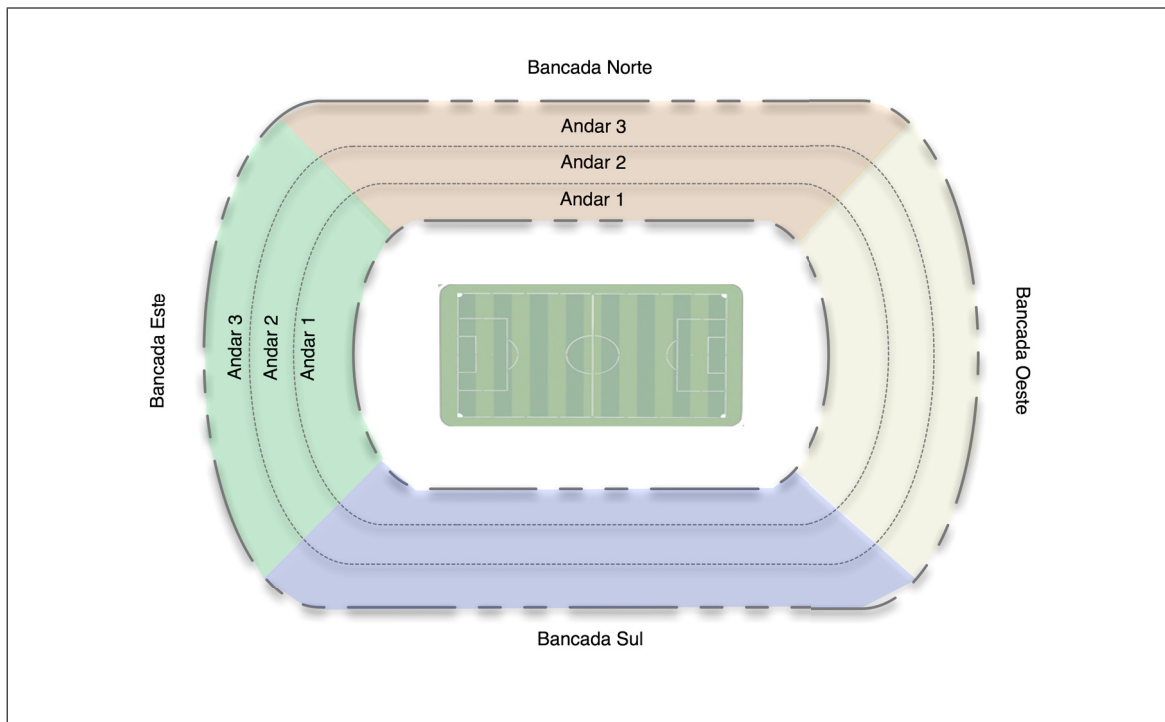


Figura 5.4: Exemplo de um estádio.

Na listagem 5.1 apresenta-se o exemplo do ficheiro de configuração, da bancada Norte, do estádio representado na figura 5.4.

```

1  #Properties File to the Test Application
2  FLOORS=FLOOR1,FLOOR2,FLOOR3
3  FLOOR3=SECTION1,SECTION2
4  FLOOR2=SECTION3,SECTION4
5  FLOOR1=SECTION5,SECTION6
6  SECTION1=ROW1,ROW2
7  SECTION2=ROW3,ROW4
8  SECTION3=ROW5,ROW6
9  SECTION4=ROW7,ROW8
10 SECTION5=ROW9,ROW10
11 SECTION6=ROW11,ROW12
12 ROW1=1,2,3,4,5,6,7,8
13 ROW2=1,2,3,4,5,6,7,8
14 ROW3=1,2,3,4,5,6,7,8
15 ROW4=1,2,3,4,5,6,7,8
16 ROW5=1,2,X,4,5,X,7,8
17 ROW6=1,2,X,4,5,X,7,8

```

```

18 ROW7=1,2,X,4,5,X,7,8
19 ROW8=1,2,X,4,5,X,7,8
20 ROW9=1,X,3,X,5,X,7,X
21 ROW10=1,X,3,X,5,X,7,X
22 ROW11=1,X,3,X,5,X,7,X
23 ROW12=1,X,3,X,5,X,7,X

```

Listagem 5.1: Exemplo de um ficheiro de configuração de uma bancada.

Com as propriedades dos locais contidas em documentos de texto, é possível “transformar” um local numa matriz. A figura 5.5 mostra o resultado da transformação do ficheiro de configurações apresentado anteriormente, numa matriz, permitindo visualizar a configuração dos três andares da bancada (*Floor 1*, *Floor 2* e *Floor 3*)².

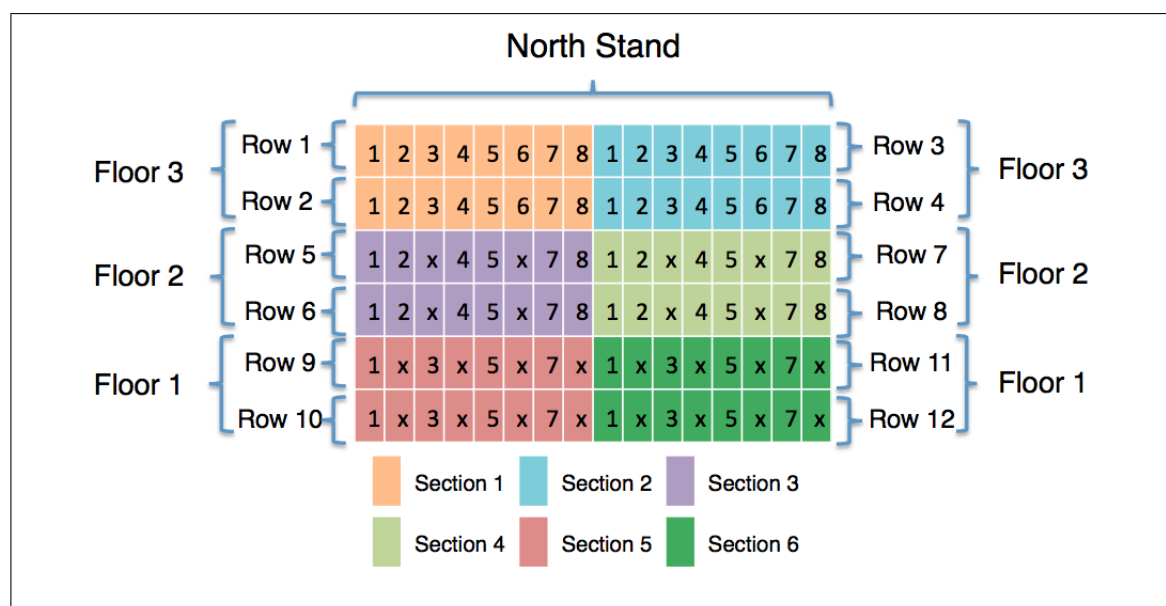


Figura 5.5: Matriz de mapeamento do estádio.

Leitura do ficheiro de configuração

Para mapear um local, é necessário efetuar a leitura do ficheiro e o respetivo carregamento de dados. Pode ser utilizada qualquer nomenclatura para a identificação das configurações de um local. No entanto, é necessário que este apresente

² Ao longo deste documento adota-se a terminologia inglesa na designação de alguns elementos do sistema uma vez que, por conveniência, na implementação esses elementos surgem na língua inglesa.

um formato coerente. No caso do exemplo do estádio apresentado anteriormente, que possui três andares identificados como “FLOOR1”, “FLOOR2” e “FLOOR3”, é necessário que exista a configuração de cada um desses andares da seguinte forma: “FLOOR1=SECTION5,SECTION6”.

Para a leitura do ficheiro que se encontra armazenado localmente e para o controlo dos índices da matriz (currentRow e currentColumn), utiliza-se em conjunto os algoritmos 1 e 2. O algoritmo 1 começa por ler o conteúdo do ficheiro correspondente ao local em questão. De seguida, obtém-se a lista de bancadas do estádio. A identificação de cada bancada, será utilizada para carregar o ficheiro que contém a respetiva configuração.

Algoritmo 1 Leitura do ficheiro de configuração.

```

PropFile ← stadiumName;
AbrirFicheiro(PropFile);
StandsArray[] ← Obter(“Stands”);
for i = 0 até StandsArray.length do
    AbrirFicheiro(StandsArray[i]);
    FloorsArray[] ← Obter(“Floors”);
    for j = 0 até FloorsArray.length do
        SectionsArray[] ← Obter(“Sections”);
        for k = 0 até SectionsArray.length do
            RowsArray[] ← Obter(“Rows”);
            for l = 0 até RowsArray.length do
                SeatsArray[] ← Obter(“Seats”);
                for m = 0 até SeatsArray.length do
                    if SeatsArray[m] != “X” then
                        SeatsMatrix[currentRow][currentColumn] ← SeatsArray[m];
                    end if
                end for
            end for
        end for
    end for
end for

```

Para controlar a posição da matriz utiliza-se o algoritmo 2. O algoritmo implementado, realiza os testes necessários para que as células da matriz se mantenham coerentes com os lugares do local. Este algoritmo começa por verificar se o lugar que se encontra em processamento é o último lugar da fila. Caso seja o último lugar da fila, verifica se a atual fila é a última da secção.

Algoritmo 2 Controlo da posição na matriz.

```

if  $currentSeat + 1 == SeatsArray.length$  then
    //Último lugar da fila;
    if  $currentLine + 1 == RowsArray.length$  then
        //Última fila da secção;
        if  $currentSection + 1 == SectionsArray.length$  then
            //Última secção do andar;
            if  $currentFloor + 1 == FloorsArray.length$  then
                //Último andar da bancada;
                if  $currentStand + 1 == StandsArray.length$  then
                    //Última bancada do estádio;
                    //Fim da processamento do estádio;
                else
                    //Não é a última bancada do estádio;
                     $currentColumn \leftarrow currentColumn + 1$ ;
                     $currentRow \leftarrow 0$ ;
                     $StartingColumnOfTheStand \leftarrow currentColumn$ ;
                     $StartingColumnOfTheSection \leftarrow currentColumn$ ;
                     $StartingRowOfTheFloor \leftarrow 0$ ;
                     $StartingColumnOfTheStand \leftarrow currentColumn$ ;
                end if
            else
                //Não é o último andar da bancada;
                 $currentRow \leftarrow currentRow + 1$ ;
                 $currentColumn \leftarrow StartingColumnOfTheStand$ ;
                 $StartingRowOfTheFloor \leftarrow currentRow$ ;
                 $StartingColumnOfTheSection \leftarrow currentColumn$ ;
            end if
        else
            //Não é a última secção do andar;
             $currentColumn \leftarrow currentColumn + 1$ ;
             $currentRow \leftarrow StartingRowOfTheFloor$ ;
             $StartingColumnOfTheSection \leftarrow currentColumn$ ;
        end if
    else
        //Não é a última fila da secção;
         $currentRow \leftarrow currentRow + 1$ ;
         $currentColumn \leftarrow StartingColumnOfTheSection$ ;
    end if
else
    //Não é o último lugar da fila;
     $currentColumn \leftarrow currentColumn + 1$ ;
end if

```

5.3.2 Módulo de localização

Para um utilizador participar num evento é necessário que o sistema possua a informação da sua localização. Assim, considera-se uma abordagem onde os dispositivos podem ser localizados de forma explícita ou estimada. De forma explícita, cabe ao utilizador indicar a sua posição no estádio. Em alternativa, considera-se a utilização do Bluetooth para descobrir os seus vizinhos, e com base nessa informação, estimar a sua localização.

Localização explícita

O método de localização explícita é baseado num conjunto de perguntas para determinar a localização do dispositivo. No início do sistema, o dispositivo recebe uma lista de locais. Ao conhecer a sua localização, o utilizador seleciona um desses locais. O local escolhido pelo utilizador é enviado ao servidor que, por sua vez, realiza uma série de “perguntas” para obter a localização do utilizador.

No final da interação, o dispositivo móvel envia uma mensagem ao servidor que, além da identificação do dispositivo, contém os parâmetros que identificam o lugar escolhido pelo utilizador. Sendo que, a identificação dos dispositivos é realizada através do International Mobile Equipment Identity (IMEI), e a identificação do lugar, no caso do exemplo que se ilustra na figura 5.4, é realizada através da bancada, do andar, da secção, da fila e do lugar.

Localização estimada

O método de localização desenvolvido no âmbito desta dissertação, tem como principal objetivo localizar os dispositivos móveis possuidores de Bluetooth. Com a utilização do Bluetooth é possível obter a localização de alguém que se encontre próximo e, com essa informação, o servidor pode atribuir um lugar que não se encontre ocupado. Apesar da precisão de localização se encontrar fora do âmbito desta dissertação, o erro máximo é o alcance rádio dos dispositivos Bluetooth (aproximadamente 10 metros na maioria dos casos).

A figura 5.6 ilustra o método de localização proposto neste trabalho. Analisando a figura verifica-se que o dispositivo d_a pretende que o sistema estime a sua localização. Assim, procura por dispositivos ao seu alcance e, no final da fase de descoberta, com o objetivo de obter informações de localização, tenta estabelecer ligação com os

dispositivos encontrados. Essa informação, quando obtida com sucesso, é enviada ao servidor que, por sua vez, determina e informa o dispositivo da sua localização.

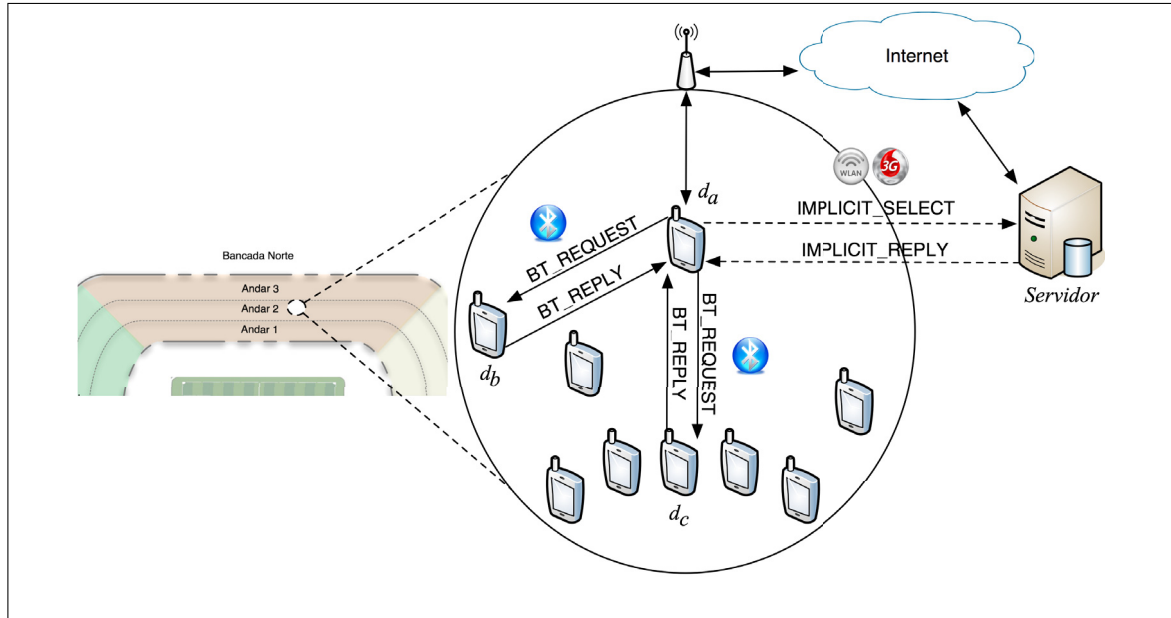


Figura 5.6: Localização de um dispositivo de forma estimada.

5.3.3 Módulo de sincronização

Um protocolo de sincronização simples é suficiente para determinar a ordem total do acontecimento dos eventos. A compensação dos diferentes relógios é realizada utilizando a hora do servidor como referência. Sugere-se que cada dispositivo seja sincronizado no momento que toma a iniciativa de participar num evento. O protocolo entre um dispositivo móvel e o servidor consiste nas seguintes fases:

1. O dispositivo d_a transmite a sua hora atual ($localTime(d_a)$) para o servidor;
2. O servidor recebe o hora local ($localTime(d_a)$) e calcula a diferença ($diff_{d_a}$) entre a sua hora atual e a hora enviada pelo dispositivo;
3. O servidor obtém a informação do local onde o dispositivo d_a se encontra ($matrixColumn_{d_a}$) e, através da equação 5.1, calcula os tempos de ocorrência

para todas as repetições (i) do evento x ;

$$\begin{aligned}
 \forall i \in event(x), \\
 totalDelay_i &= (matrixColumn_{d_a} \times delay_x) + (i \times maxColumns_{d_a} \times delay_x) \\
 eventTime_i &= (beginTime_x + totalDelay_i) - diff_{d_a}
 \end{aligned}
 \tag{5.1}$$

4. O dispositivo d_a recebe a informação do evento e termina o processo de sincronização. Caso contrário, o dispositivo d_a recomeça o protocolo de sincronização.

Depois de uma execução bem sucedida do protocolo, o dispositivo possui a informação necessária para participar nas várias ocorrências de um evento. Em caso de sucesso, o protocolo necessita apenas de duas mensagens para sincronizar o dispositivo com o servidor. O protocolo não tem em conta a variação no atraso das mensagens. A estimativa é imprecisa, mas é suficiente. É bastante simples e leve e, portanto, adequado para utilização no contexto do trabalho proposto, onde é necessário sincronizar inúmeros dispositivos.

5.4 Protocolo de comunicação

Os intervenientes do SyncEvent devem trocar mensagens através da rede, verificar a validade e disponibilizar o conteúdo das mensagens aos restantes módulos do sistema. Nesta secção é descrito o protocolo para comunicação entre os dispositivos móveis e o servidor. Este protocolo é bastante simples e repete-se em cada dispositivo que deseje participar num determinado evento.

O protocolo contém duas fases: a fase de localização e a fase de participação num evento. Na fase de localização cada dispositivo decide, de forma independente, se pretende ser localizado de forma explícita ou estimada. Um dispositivo que pretenda ser localizado de forma estimada, procura e tenta estabelecer uma ligação de dados com os dispositivos ao seu alcance. No caso de uma ligação estabelecida com sucesso, tenta obter a localização do dispositivo. No entanto, um dispositivo pode não conseguir encontrar, ou simplesmente não possuir, dispositivos ao seu alcance. Portanto, um dispositivo que não seja capaz de obter a informação de localização de um dos

seus vizinhos, não pode ser localizado de forma estimada. Da mesma forma, um dispositivo que não consiga estabelecer ligação com o servidor, não pode participar no evento. Quando um utilizador pretende participar num evento, informa o servidor da sua intenção. Este recebe os pedidos dos utilizadores e, tendo em conta a localização dos dispositivos, responde com a informação dos eventos.

Para uma melhor compreensão dos mecanismos de localização e de participação em eventos, apresentam-se de seguida alguns detalhes sobre esses processos.

5.4.1 Localização explícita de dispositivos

Quando um determinado utilizador inicia a aplicação SyncEvent e pretende ser localizado de forma explícita, é trocada uma sequência de mensagens entre o dispositivo móvel e o servidor. As mensagens trocadas nesse instante inicial, têm o objetivo de obter a localização geográfica dos dispositivos móveis para, posteriormente, ser possível sincronizar os eventos. A partir da figura 5.7 identificam-se as mensagens necessárias para que, no cenário do estádio apresentado na figura 5.4, seja possível localizar um dispositivo.

A figura 5.7 ilustra o processo de localização explícita de um dispositivo no estádio apresentado na figura 5.4. Lembra-se que, no contexto do estádio apresentado anteriormente, os lugares ocupados pelos utilizadores são identificados através do seguinte conjunto de parâmetros: bancada, andar, secção, fila e lugar. Desta forma, um utilizador que deseje ser localizado de forma explícita, deve conhecer e comunicar ao servidor o valor desses parâmetros.

O processo de localização explícita é baseado no modelo de pedido/resposta³. Neste caso em particular, o servidor possui as propriedades de cada local. Desta forma, o dispositivo deve pedir a lista de atributos para que o utilizador selecione o seu lugar. Através da análise da figura 5.7, verifica-se que todas as mensagens utilizam o seguinte método: o cliente repete periodicamente o pedido até que receba o resultado pretendido ou, até que seja atingido o número máximo de tentativas (*maxAttempts*).

³ No modelo pedido/resposta, um dispositivo estabelece uma ligação e envia um pedido ao servidor, o qual analisa e responde.

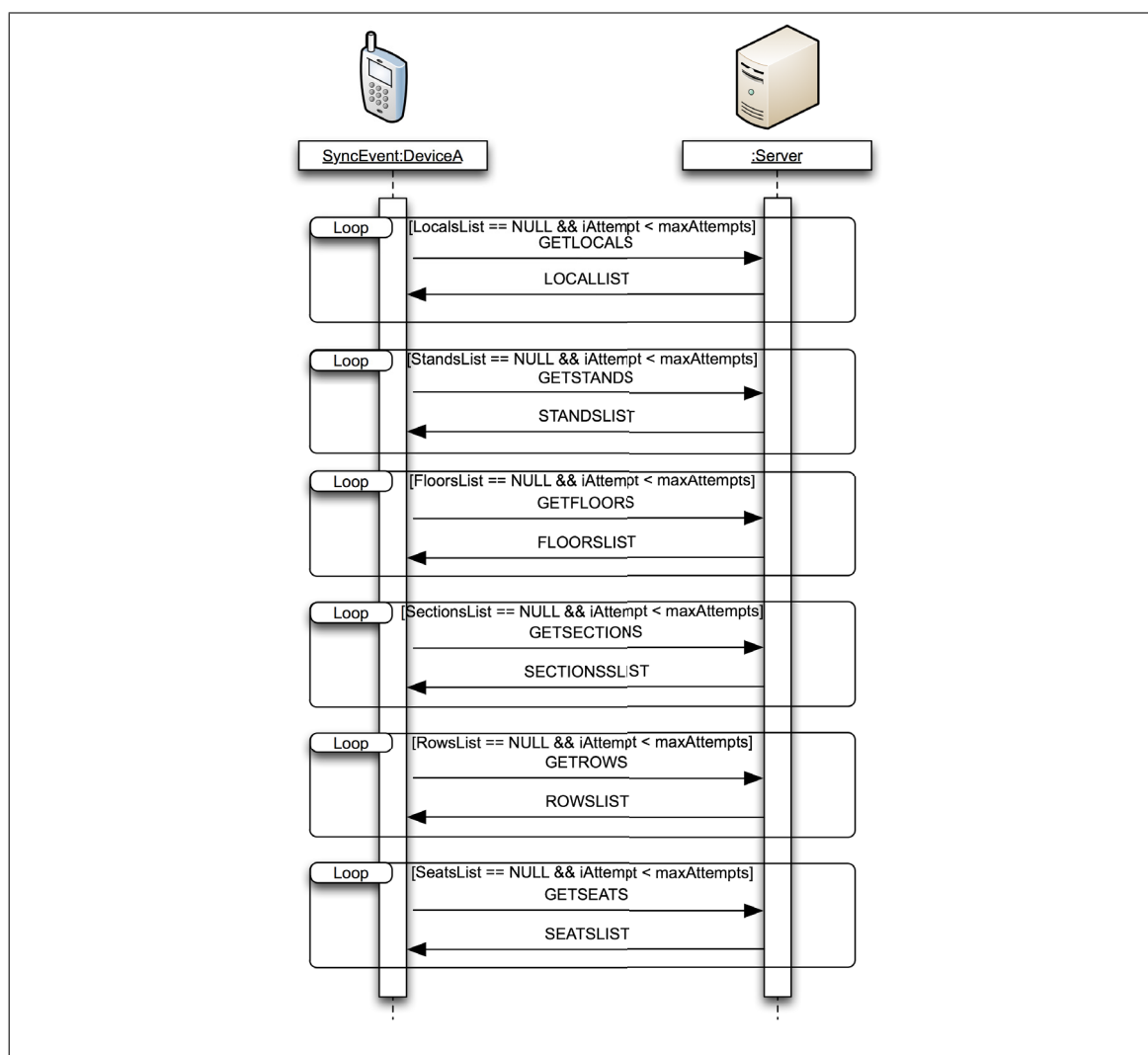


Figura 5.7: Localização explícita de dispositivos.

5.4.2 Localização implícita de dispositivos

Com a utilização do Bluetooth é possível descobrir os vizinhos e obter a informação de localização destes. Com essa informação, é possível estimar a localização do dispositivo num determinado local. Para que o servidor estime a localização de um determinado dispositivo, este procura e tenta estabelecer ligação com dispositivos Bluetooth ao seu alcance. A figura 5.8 ilustra o diagrama de sequência para a localização implícita de dispositivos.

Após o estabelecimento de ligação com um dos dispositivos ao alcance, o dispositivo que pretende ser localizado automaticamente ($Device_A$), pede a informação de localização do seu vizinho ($Device_B$). Caso a localização do vizinho seja recebida

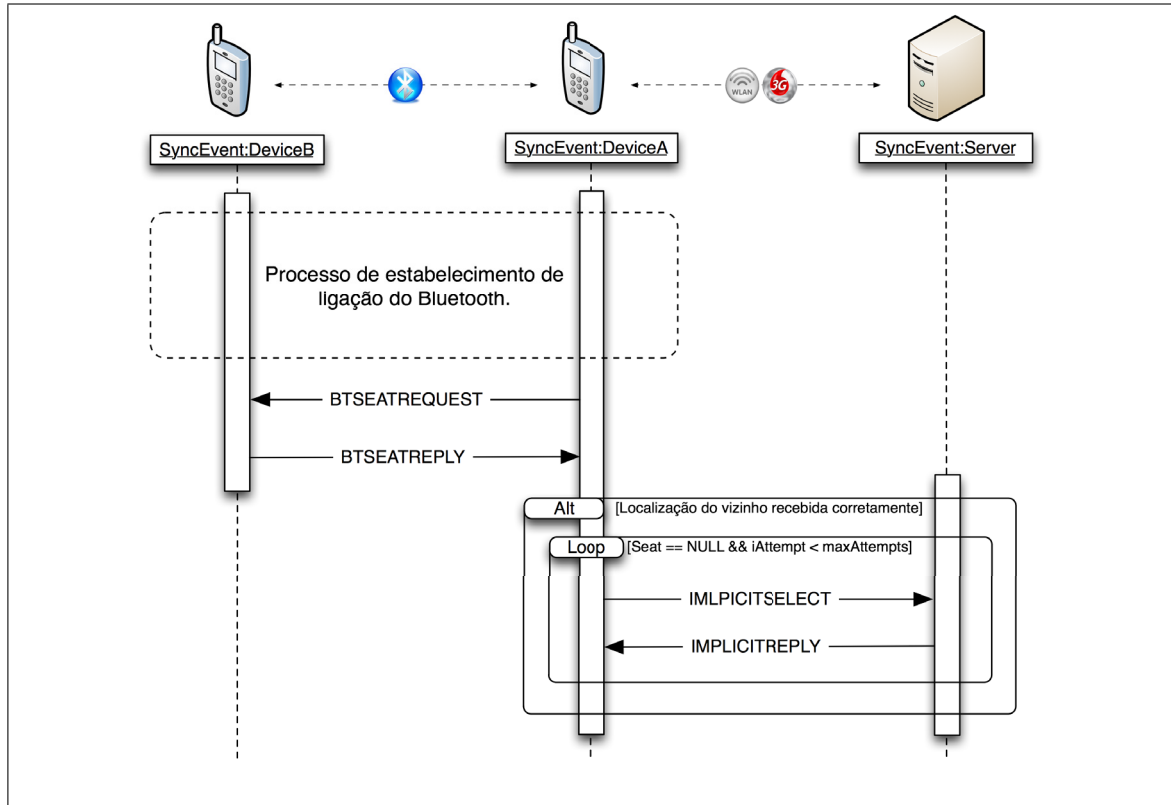


Figura 5.8: Localização implícita de dispositivos.

corretamente, o dispositivo *Device_A* envia essa informação ao servidor (*Server*) e aguarda pela resposta. Neste processo, utiliza-se o método do processo de localização explícita (secção 5.4.1), onde o dispositivo repete o pedido, em intervalos regulares, até rececionar a resposta do servidor ou, até que seja atingido o número máximo de tentativas (*maxAttempts*).

5.4.3 Participação num evento

A participação num evento encontra-se dependente da vontade do utilizador em participar. Depois do processo de localização concluído com sucesso, o utilizador tem a possibilidade de participar num evento que se encontre disponível no local onde se encontra. Assim, quando desejar, comunica ao servidor a sua intenção. Observando a figura 5.9 e admitindo que o processo de localização foi concluído com sucesso, pode-se constatar que a participação num evento encontra-se apenas dependente da troca de duas mensagens (*EVENTJOIN* e *EVENTREPLY*).

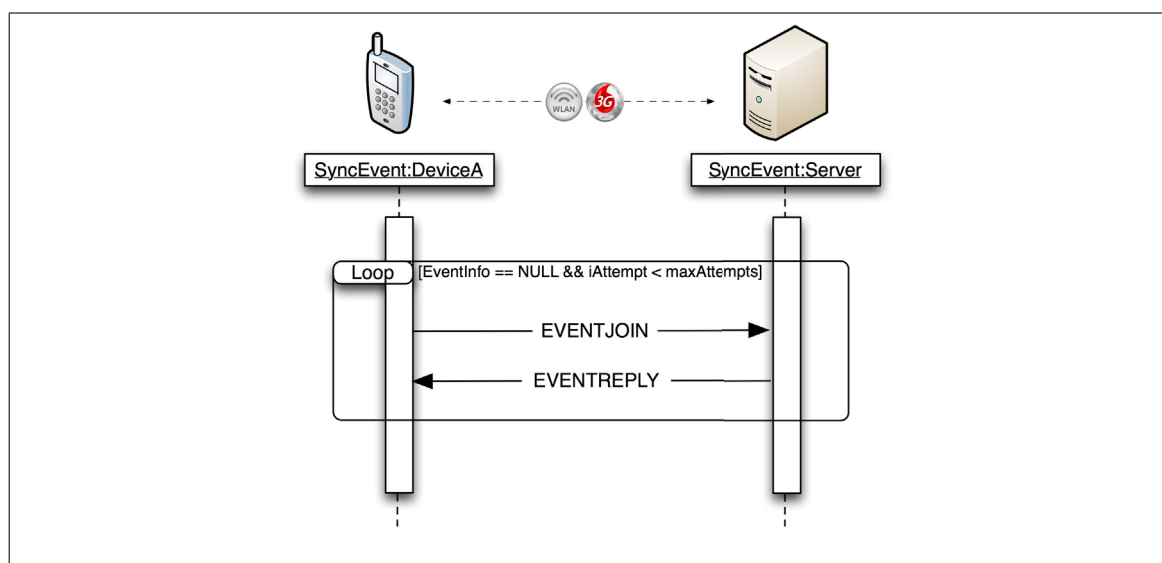


Figura 5.9: Participação num evento.

5.4.4 Formato das mensagens

A comunicação entre os intervenientes do sistema SyncEvent é baseada nas mensagens definidas nas próximas secções. Contudo, antes da possibilidade de ser transmitida qualquer mensagem, deve ser realizado um processo de estabelecimento de ligação.

Formato geral

A figura 5.10 ilustra o formato geral das mensagens transferidas entre o cliente e o servidor do SyncEvent. Cada mensagem é composta pôr três parâmetros: *MessageType*, *LocationMessage* e *EventMessage*.

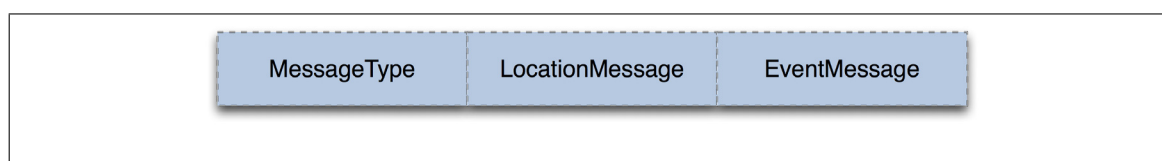


Figura 5.10: Formato geral das mensagens trocadas entre os intervenientes do SyncEvent.

Os parâmetros *LocationMessage* e *EventMessage* correspondem respetivamente às mensagens de localização e de eventos. O parâmetro *MessageType* possui a identificação do tipo de mensagem enviada. Na tabela 5.1 encontram-se os tipos de mensagens definidos no protocolo do SyncEvent.

Tabela 5.1: Tipos de mensagens definidos para a comunicação entre os intervenientes do sistema.

Valor	Mensagem	Valor	Mensagem
0	LOCALS_REQUEST	10	SEATS_REQUEST
1	LOCALS_REPLY	11	SEATS_REPLY
2	STANDS_REQUEST	12	IMP_SELECT
3	STANDS_REPLY	13	IMP_REPLY
4	FLOORS_REQUEST	14	SELECT_REPLY
5	FLOORS_REPLY	15	SEAT_SELECT
6	SECTIONS_REQUEST	16	BT_SEAT_REQ
7	SECTIONS_REPLY	17	BT_SEAT_REP
8	ROWS_REQUEST	18	EVENT_TJOIN
9	ROWS_REPLY	19	EVENT_REPLY

As diferentes mensagens enviadas para o servidor são consequência das decisões tomadas por cada utilizador. Para a localização do utilizador são utilizadas as mensagens numeradas de 0 a 17, sendo as 2 restantes utilizadas quando o utilizador decide participar num evento.

Mensagem *LocationMessage*

A mensagem de localização, ou *LocationMessage* como se encontra denominada na figura 5.11, consiste no código de identificação do dispositivo (*DEVICE_ID*), e num conjunto de parâmetros utilizados para a localização do dispositivo. Os campos desta mensagem possuem um comprimento variável, podendo mesmo encontrar-se sem conteúdo.

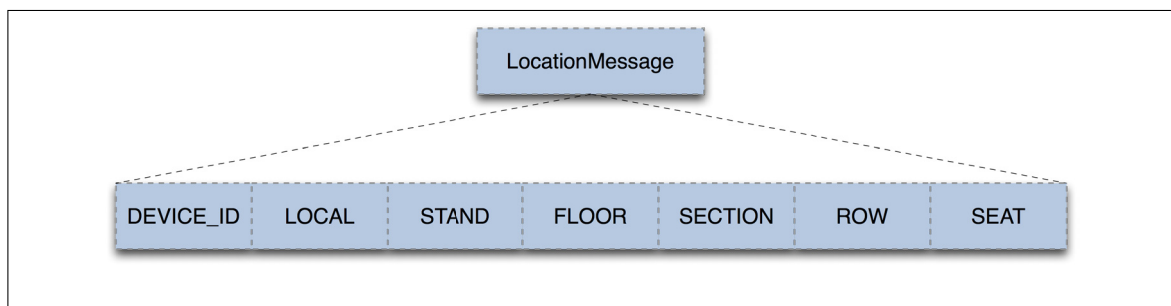


Figura 5.11: Formato das mensagens trocadas entre os intervenientes do SyncEvent no processo de localização.

Mensagem *EventMessage*

A mensagem *EventMessage* é usada para informações relacionadas com eventos. Esta mensagem possui, entre outros parâmetros, o relógio local do dispositivo que, em conjunto com as informações de localização, permite ao servidor realizar os cálculos necessários para que os eventos ocorram de forma sincronizada.

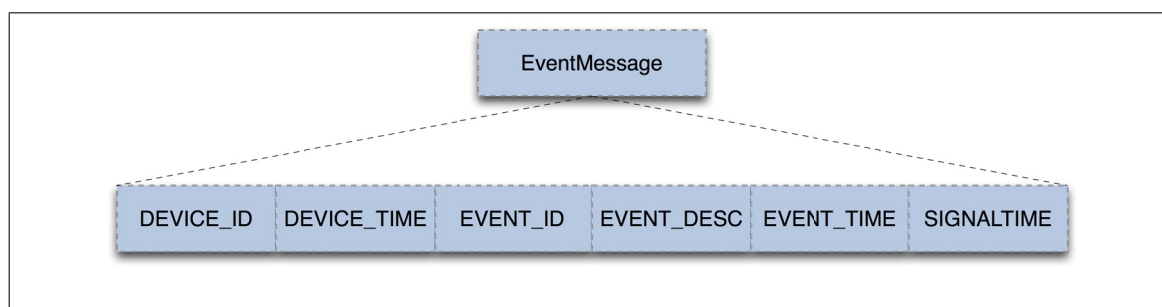


Figura 5.12: Formato das mensagens trocadas entre os intervenientes do SyncEvent no processo de participação em eventos.

5.5 Conclusões

Durante o decorrer deste capítulo foram identificadas várias funcionalidades para o SyncEvent: mapeamento de locais; localização dos dispositivos; difusão de eventos e sincronização dos eventos. Para desempenhar essas funcionalidades, foi desenvolvida uma arquitetura que permite aos utilizadores comunicar com o servidor através de uma infraestrutura WiFi presente no local, ou diretamente através de uma outra tecnologia de acesso à Internet presente nos dispositivos móveis dos utilizadores.

Com as configurações dos locais num documento de texto consegue-se representar os locais através de uma matriz. Cada célula da matriz representa um lugar de um determinado local.

Em relação à localização dos dispositivos, foi adotada uma abordagem onde os utilizadores podem ser localizados de forma explícita ou estimada. De forma explícita, os utilizadores tomam a iniciativa de informar o servidor da sua localização. No entanto, com a utilização do Bluetooth, é possível obter a informação de localização dos vizinhos e, com essa informação, estimar a localização do dispositivo.

Os dispositivos seguem um protocolo simples, necessitando apenas de duas mensagens, para sincronizar os eventos. O processo de sincronização é realizado no momento em que o utilizador resolve participar num evento.

Capítulo 6

Implementação

Este capítulo realiza a descrição da implementação do servidor e da aplicação para os dispositivos móveis. Inicialmente é descrito o cenário de implementação e, de seguida, são discutidas e justificadas as opções tomadas. No final do capítulo apresentam-se os resultados obtidos.

6.1 Descrição do cenário

A participação num evento está dependente da vontade do utilizador em participar. A aplicação deste sistema possui um sem número de eventos, apenas é necessária alguma criatividade para despertar o interesse dos visitantes.

O evento mais simples, e provavelmente o principal evento realizado pelos espectadores de um estádio, é conhecido por “onda mexicana” (figura 6.1). A “onda mexicana” ou *La Ola*, como ficou conhecida durante o campeonato do Mundo realizado no México em 1986, é uma onda humana, que percorre as bancadas de um estádio [Farkas et al., 2002]. Para produzir o efeito visual chamado de “onda”, as pessoas levantam os braços de forma coordenada, ao longo de todo o estádio.

Devido à relativa simplicidade do efeito de “onda”, o primeiro evento disponibilizado pelo SyncEvent é uma reprodução desse comportamento coletivo. A diferença para o tradicional modelo de braços no ar é que agora o efeito de “onda” é produzido acendendo, de forma coordenada, a iluminação do visor dos dispositivos móveis.



Figura 6.1: Exemplo de uma onda Mexicana ou *La Ola* [Farkas et al., 2002].

Na listagem 6.1, apresenta-se um exemplo de configuração de um evento.

1	EventId = 1;	# Identificação do evento.
2	EventDesc = SingleWave;	# Descrição do evento.
3	BeginTime = 11:58:00:000;	# Hora inicial do evento.
4	SignalTime = 00:00:01:000;	# Duração do evento em cada dispositivo.
5	Delay = 00:00:00:250;	# Atraso do evento entre vizinhos.
6	MaxRepeats = 5;	# Número de ocorrências do evento.

Listagem 6.1: Exemplo de configuração de um evento.

Analisando a listagem 6.1 verifica-se que, para a configuração de um evento, são necessários seis parâmetros. A configuração possui, entre outros parâmetros, a hora inicial do evento (*BeginTime*), que em conjunto com a duração do sinal em cada dispositivo (*SignalTime*) e com o atraso (*Delay*), permite ao servidor sincronizar a ocorrência dos eventos nos dispositivos móveis.

6.2 Escolha do sistema operativo

A partir do estudo na área dos dispositivos móveis, verifica-se que os *smartphones* são a principal tendência no mercado de telecomunicações. As principais características de um *smartphone* encontram-se em praticamente todos os modelos existentes no mercado. No entanto, uma das diferenças entre os diversos modelos é o sistema operativo.

Android

Para a criação de eventos sincronizados em dispositivos móveis fisicamente próximos foi desenvolvida uma aplicação direcionada a *smartphones* com o sistema operativo Android da Google. Apesar de não existir um sistema operativo predominante, todos eles possuem diferentes características que devem ser consideradas no momento de seleção. O sistema operativo Android, além de possuir uma elevada quota de mercado, tem a vantagem de ser *open source*.

Ao contrário de outros sistemas operativos, o desenvolvimento de aplicações para Android é completamente livre. O Android disponibiliza gratuitamente uma ampla gama de bibliotecas e de ferramentas para o desenvolvimento de aplicações. Além disso, a submissão de aplicações, na loja de aplicações *online* da Google, acontece de imediato. Depois de terminada a aplicação não é necessário aguardar pela aprovação da Google para que a aplicação fique disponível para os utilizadores.

Versões do Android

O Android tem passado por uma sucessão de alterações desde o seu lançamento inicial. Em cada atualização são corrigidos erros de versões anteriores e são adicionadas novas funcionalidades. A tabela 6.1 mostra as versões do sistema operativo Android lançadas até à data da realização desta dissertação.

Tabela 6.1: Versões do sistema operativo Android [Android, 2011].

Versão	Nome de Código	Lançamento	Distribuição
1.5	Cupcake	abril de 2009	1.1%
1.6	Donut	setembro de 2009	1.4%
2.1	Eclair	outubro de 2009	11.7%
2.2	Froyo	maio de 2010	45.3%
2.3.x	Gingerbread	dezembro de 2010	38.7%
3.x	Honeycomb	janeiro de 2011 ¹	1.8%

Analisando a tabela 6.1 verifica-se a quantidade de dispositivos abrangida pela aplicação desenvolvida. A coluna de distribuição indica a percentagem total de dispositivos, presentemente ativos, compatíveis com uma determinada versão do Android.

¹ Versão lançada especialmente para *tablets*.

As APIs mais recentes disponibilizam suporte para as versões anteriores [Android, 2011]. Assim, caso uma aplicação seja desenvolvida para a versão mais antiga, será suportada pela totalidade dos dispositivos Android. Contudo, as versões mais recentes possuem funcionalidades inexistentes em versões anteriores.

A aplicação do dispositivo móvel foi desenvolvida utilizando a API da versão 2.1. Deste modo, é possível obter funcionalidades recentes e, segundo a distribuição apresentada na tabela 6.1, conseguir que a aplicação seja suportada por 97.5% dos dispositivos móveis com Android.

Arquitetura

Na figura 6.2 apresenta-se a arquitetura global do Android. Esta arquitetura é construída a partir do agrupamento de um conjunto de camadas.

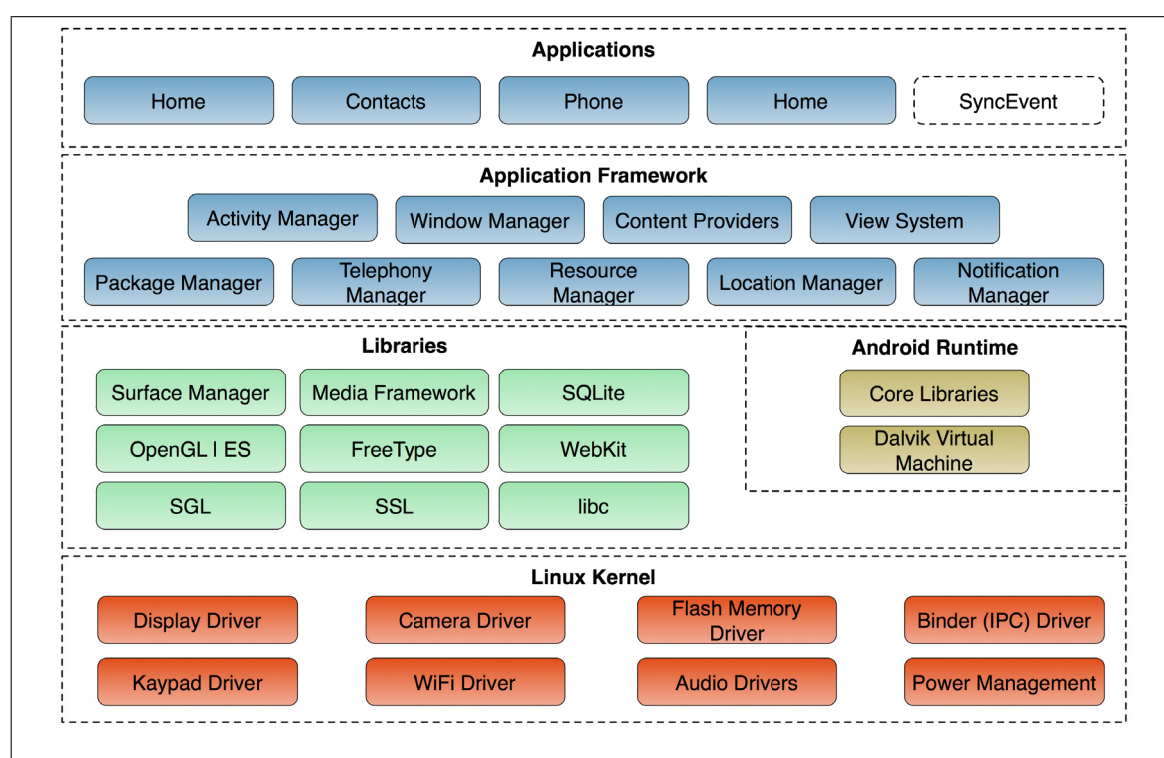


Figura 6.2: Arquitetura global do Android [Android, 2011].

Analisando a figura 6.2 verifica-se que a camada inferior é designada por *Linux Kernel*. Esta camada, baseada na versão 2.6 do Linux, possui funcionalidades como gestão de memória e gestão de processos. Em cima do *kernel*, o Android disponibiliza um conjunto de bibliotecas nativas desenvolvidas em C/C++. Ao mesmo

nível, encontram-se implementadas grande parte das funcionalidades do Java. Cada aplicação é executada num processo independente, com uma instância autónoma da máquina virtual Dalvik. Ainda em cima destas camadas encontram-se os elementos utilizados no desenvolvimento de aplicações.

Componentes

Segundo o site oficial do Android, os componentes são os elementos essenciais de uma aplicação. Os componentes possuem diversas funcionalidades e diferentes estados, determinando o comportamento das aplicações. A seguir apresenta-se uma breve descrição dos componentes definidos pelo Android [Android, 2011]:

Activities - Uma *activity*, ou atividade na língua Portuguesa, representa a componente de interface com o utilizador. Habitualmente, uma aplicação possui diversas atividades que, embora sejam autónomas, trabalham em conjunto com o objetivo de disponibilizar uma experiência de utilização coerente. Normalmente, uma aplicação define uma atividade principal, apresentada ao utilizador quando este inicia a aplicação. De forma a executar outras operações, as atividades podem inicializar outras atividades.

Services - Um *Service* (ou serviço), é normalmente utilizado para executar operações mais demoradas em plano secundário. Por exemplo, um serviço pode realizar *downloads* ou reproduzir música em plano secundário. Contudo, os serviços não disponibilizam interface com o utilizador.

Content providers - Estes componentes são utilizados para armazenar e recuperar informação. Os *content providers* representam a melhor forma de partilhar informação entre as aplicações. Através destes é possível, por exemplo, armazenar a informação no sistema de ficheiros ou numa base de dados SQLite. Um dos *content providers* disponibilizado pelo Android realiza a gestão dos contactos do utilizador. Desta forma, qualquer aplicação, com as permissões necessárias, pode ler, editar e gravar, informações dos contactos (nome, morada, número de telefone, etc.).

Broadcast receivers - Um *broadcast receiver* é um componente que possui acesso a notificações de todo o sistema. O Android define diversas notificações, tais como aviso de ecrã desligado, aviso de bateria fraca, entre outras. Qualquer aplicação

pode disponibilizar notificações. Estes componentes não possuem uma interface para o utilizador, no entanto, o Android permite que sejam criadas notificações na barra de estados para informar o utilizador de um determinado acontecimento.

Ciclo de vida das atividades

Cada vez que é iniciada uma nova atividade, a anterior é suspensa. Ainda assim, a atividade é adicionada a uma lista do tipo “*last in, first out*”, proporcionando ao utilizador a possibilidade de voltar a atividades anteriores. Segundo o site oficial do Android, uma atividade possui quatro estados essenciais [Android, 2011]:

1. Quando uma atividade é iniciada diz-se que se encontra em execução ou que se está ativa.
2. Quando uma atividade perde o foco, mas ainda se encontra visível, a atividade encontra-se suspensa ou pendente. Quando uma atividade entra neste estado, apesar de manter as informações da atividade, pode ser destruída pelo sistema por falta de memória.
3. Quando uma atividade fica totalmente oculta é interrompida. Neste estado, a atividade não se encontra visível para o utilizador, no entanto, para que seja possível voltar à atividade, o sistema conserva as informações necessárias. Tal como no estado anterior, quando uma atividade se encontra neste estado, pode ser destruída pelo sistema.
4. Quando uma atividade se encontra suspensa ou interrompida, o sistema pode proceder à destruição da atividade para libertar recursos para outras aplicações. Para que seja novamente apresentada ao utilizador tem de ser reiniciada.

Interface do utilizador

Numa aplicação Android, a interface do utilizador é definida através das classes *View* e *ViewGroup* [Android, 2011]. A classe *View* é fundamental para a criação de elementos como caixas de texto e botões. A classe *ViewGroup*, por sua vez, é a base para a realização de *layouts*. Normalmente os *layouts* são concebidos em XML. Na listagem 6.2 apresenta-se parte de um dos *layouts* definidos no SyncEvent.

```
1      ...
2      <TextView android:id="@+id/TextView_TimeRemain"
3          android:layout_height="wrap_content"
4          android:textSize="16pt"
5          android:text="00:00:00"
6          android:textColor="#00C2FF"
7          android:layout_gravity="center_horizontal"
8          android:layout_width="wrap_content">
9      </TextView>
10     ...
```

Listagem 6.2: Parte da configuração de um *layout* do SyncEvent.

6.3 Escolha da linguagem de programação

Um dos pontos que deve ser discutido quando se inicia um novo projeto é a linguagem de programação a utilizar. Segundo Spinellis [2006], os principais fatores na escolha de uma linguagem de programação são a produtividade, a manutenção, a eficiência e o suporte de ferramentas de desenvolvimento.

Foi escolhida a linguagem Java tanto para o desenvolvimento da aplicação móvel como também para o desenvolvimento do servidor. A utilização desta linguagem justifica-se, não só pelo conhecimento prévio nesta linguagem de programação, mas também, e acima de tudo, devido às suas características de portabilidade e ao elevado número de bibliotecas disponíveis.

6.4 Protocolo de comunicação

Grande parte dos dispositivos móveis ainda possuem restrições em termos de bateria. Gligoric et al. [2011] referem que, um dos aspetos que merece alguma consideração para minimizar o consumo energético das aplicações, é a transferência de informação entre os dispositivos móveis e o servidor.

O *ProtoBuf*, desenvolvido pela Google, pretende minimizar o problema do elevado número de pedidos e respostas de e para os servidores [Gligoric et al., 2011]. Este protocolo realiza uma codificação binária de forma a tornar mais compacta a

informação transmitida. Segundo o site oficial do *ProtoBuf*, este mecanismo suporta diferentes linguagens de programação, incluindo Java, C++ e Python.

Para realizar a serialização¹ da informação, é necessário que as estruturas de dados sejam descritas e compiladas a partir de um documento “.proto”. O processo de compilação gera um conjunto de métodos na linguagem de programação escolhida. Findo o processo de compilação, os métodos disponibilizados podem ser utilizados para escrita e leitura de dados.

Em apêndice A, encontra-se o conteúdo do documento “.proto” que define as estruturas de dados utilizadas para a comunicação entre os dispositivos móveis e o servidor do SyncEvent.

6.5 Elementos do sistema

Ao longo desta secção descreve-se a relação entre os principais elementos do sistema. A relação é exposta através do diagrama de classes de cada subsistema. A criação destes diagramas, resultou de um processo onde se identificaram os elementos mais relevantes no contexto do trabalho proposto.

Com estes diagramas, pretende-se disponibilizar uma perceção simples das classes mais relevantes para o SyncEvent. Para esse efeito, estes diagramas apresentam apenas a interação entre as principais classes do sistema desenvolvido.

6.5.1 Elementos do servidor

Na figura 6.3 apresenta-se, de forma simplificada, a estrutura global do servidor do SyncEvent e a interação entre as diversas classes da aplicação.

As classes fundamentais que se encontram na figura 6.3 descrevem-se da seguinte forma:

SyncEventServer - Classe de inicialização do servidor do SyncEvent. Esta classe é responsável pela criação de uma interface gráfica, onde são registados os acontecimentos mais relevantes do sistema, e pela criação de um processo que se mantém, de forma passiva, à “espera” de mensagens num *socket* UDP. Este

¹ A serialização é o processo de converter um conjunto de dados num conjunto de bytes sendo possível, posteriormente, obter a informação (<http://pt.wikipedia.org/wiki/SerIALIZAÇÃO>).

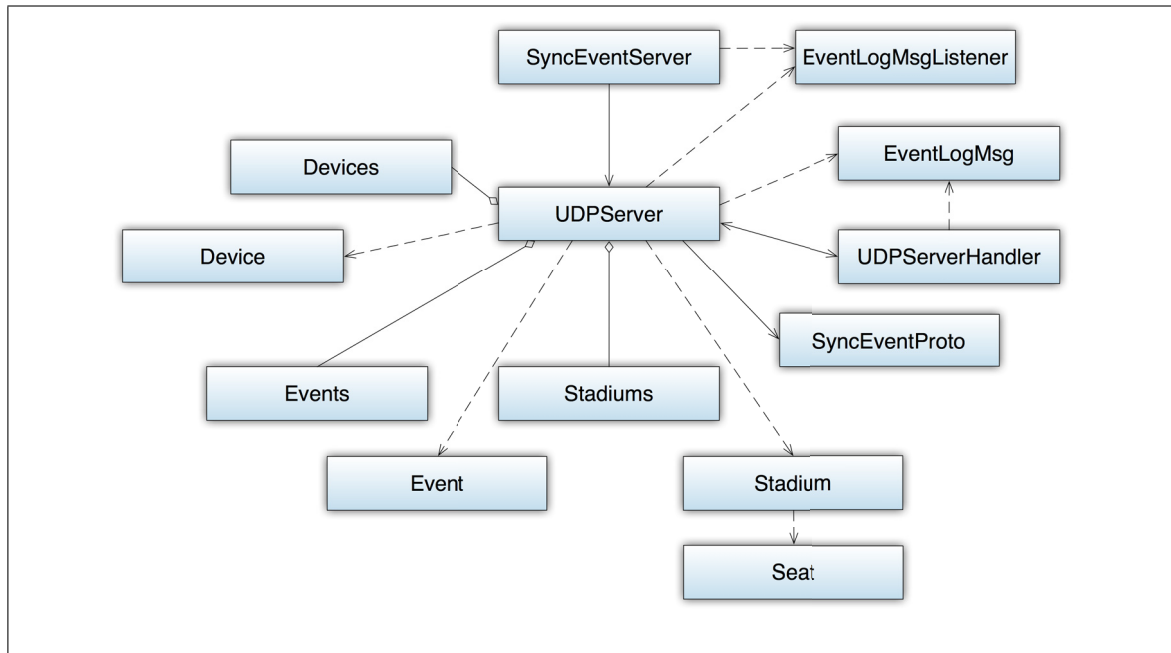


Figura 6.3: Simplificação das relações entre as classes do servidor.

processo permanece em funcionamento durante toda a atividade do servidor, lidando com a recepção de várias mensagens.

UDPServer - Classe responsável por todo o processo de comunicação. Esta classe define a porta onde o servidor se encontra à “espera” de novas mensagens. As mensagens dos diferentes dispositivos são independentes e são transmitidas através do protocolo de transporte UDP. Quando é recebida uma nova mensagem, é criada uma nova *thread* que lida com esse pedido. Esta classe implementa os métodos necessários para a gestão de todos os elementos do sistema. Assim, encontra-se dependente de praticamente todas as classes do servidor.

UDPServerHandler - Classe que executa o tratamento das mensagens recebidas. Além de implementar os métodos necessários para lidar com as mensagens recebidas pelos utilizadores do sistema, é ainda responsável pela construção das respostas enviadas aos utilizadores. Esta classe contém ainda um método que, no momento de receção de um pedido, verifica se o dispositivo já existe na lista de dispositivos. Caso não exista, insere e atualiza a hora do último pedido. Caso já exista, apenas atualiza o valor do último pedido.

Device, Event e Stadium - As classes *Device*, *Event* e *Stadium* representam res-

petivamente a descrição dos objetos dispositivo, evento e estádio. Todas essas classes descrevem as propriedades e comportamentos dos objetos que permitem instanciar.

6.5.2 Elementos do dispositivo móvel

Na figura 6.4 apresenta-se, de forma simplificada, a estrutura global da aplicação para os dispositivos móveis do SyncEvent.

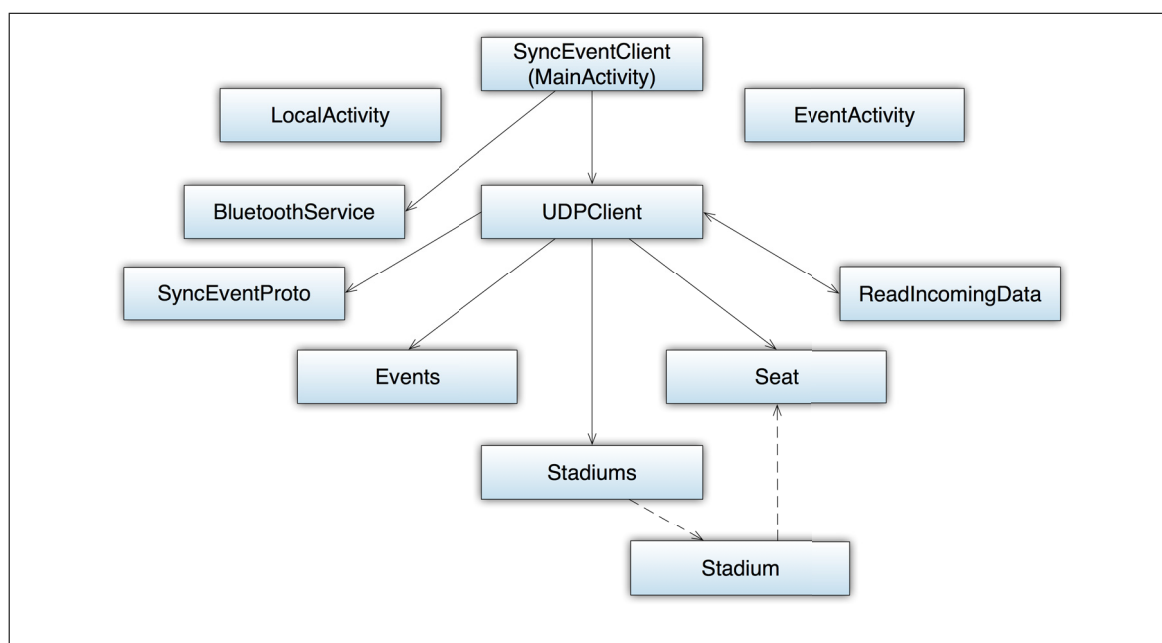


Figura 6.4: Simplificação das relações entre as classes da aplicação móvel.

As classes fundamentais do dispositivo móvel descrevem-se da seguinte forma:

SyncEventClient - A classe *SyncEventClient* corresponde à atividade apresentada ao utilizador quando este inicia a aplicação. A atividade começa por, com as permissões necessárias, ativar o Bluetooth. De seguida, envia uma mensagem ao servidor com o objetivo de receber a lista dos locais disponíveis. Caso a lista de locais seja obtida com sucesso, a atividade apresenta a informação ao utilizador e aguarda que este selecione um local ou indique a intenção de ser localizado de forma estimada. Esta atividade dá início a uma outra atividade no momento em que a localização do dispositivo é obtida com sucesso.

LocalActivity - Findo o processo de localização, é criada a atividade *LocalActivity*. Esta atividade possui um *layout* que contém dois propósitos. Antes de

mais apresenta a informação da localização atual. Simultaneamente, assegura a interface graças ao qual o utilizador informa o dispositivo da intenção em participar num evento, ou de alterar a sua localização.

EventActivity - A classe designada por *EventActivity* implementa uma atividade que aguarda pelo acontecimento de um evento. Além de possuir um *layout*, responsável pela apresentação do tempo em falta para o acontecimento de um novo evento, avisa o utilizador da aproximação de um evento. Quando se aproxima um evento, é produzida vibração no dispositivo a fim de avisar o utilizador.

UDPClient - Tal como o servidor do SyncEvent, a aplicação móvel utiliza simultaneamente processos cliente e servidor. Desta forma, a classe *UDPClient* mantém um processo que se permanece à espera de novas mensagens durante o tempo de vida da aplicação. Além disso, implementa os métodos necessários para construir as mensagens que são enviadas ao servidor.

BluetoothService - Esta classe utiliza as APIs do Bluetooth disponibilizadas pelo Android. O Android disponibiliza suporte à troca de informações entre dispositivos através do Bluetooth. Esta classe permite obter os dispositivos que se encontram próximos e estabelecer ligações de dados com esses dispositivos. Esta classe é utilizada no momento em que o utilizador pretende ser detetado de forma estimada ou, quando um dos dispositivos vizinhos deseja conhecer a localização do dispositivo para ele próprio ser localizado de forma estimada.

6.6 Dispositivos móveis utilizados

O próximo passo, no processo de desenvolvimento do sistema, foi a escolha de dispositivos apropriados. Uma vez que o emulador disponibilizado pelas ferramentas de desenvolvimento do Android é bastante limitado, foi necessário realizar testes com diferentes dispositivos móveis. A aplicação desenvolvida não exige muitos recursos, sendo compatível com praticamente todos os dispositivos móveis com Android (sendo apenas necessário que estes possuam WiFi ou conectividade à Internet via rede GSM ou UMTS/HSPDA+). De seguida, apresentam-se as especificações dos dispositivos móveis utilizados para o desenvolvimento e teste da aplicação móvel do SyncEvent.

6.6.1 Samsung Galaxy S

O modelo Samsung Galaxy S (figura 6.5) é um dos modelos topo de gama com sistema operativo Android. A tabela 6.2 apresenta as suas principais características técnicas. Presentemente já se encontra no mercado o modelo Samsung Galaxy S2 que é uma evolução do modelo utilizado para testes.



Figura 6.5: Samsung i9000 Galaxy S 8GB [Samsung, 2011].

Tabela 6.2: Especificações técnicas do Samsung i9000 Galaxy S 8GB [Samsung, 2011].

Banda	GSM/EDGE (850/900/1800/1900)
Rede & Dados	3G (HSDPA 7,2 / HSUPA 5,76)
Sistema Operativo	Android 2,1 (Eclair)
Processador	1 GHz
Dimensão	122,4 x 64,2 x 9,9 mm
Ecrã	SAMOLED - WVGA (480 x 800)
Bateria	Li-Po 1500mAh
Câmara	5 MP
Bluetooth	3.0
WiFi	802,11 b/g/n
Memória RAM	512MB

6.6.2 Samsung Galaxy Mini

O Samsung Galaxy Mini (figura 6.6) é um telefone de gama média. Embora não seja tão poderoso quanto o Samsung Galaxy S, o seu custo é bastante inferior o que o torna bastante popular. A tabela 6.3 apresenta as suas principais especificações técnicas.



Figura 6.6: Samsung S5570 Galaxy Mini [Samsung, 2011].

Tabela 6.3: Especificações técnicas do Samsung S5570 Galaxy Mini [Samsung, 2011].

Banda	GSM/EDGE (850/900/1800/1900)
Rede & Dados	3G (HSDPA 7,2)
Sistema Operativo	Android 2,2
Processador	600 MHz
Dimensão	110,4 x 60,6 x 12,1 mm
Ecrã	TN - QVGA (240 x 320)
Bateria	Li-Ion 1200 mAh
Câmara	3 MP
Bluetooth	2.1
WiFi	802,11 b/g/n
Memória RAM	164 MB

6.7 Resultados obtidos

O trabalho realizado ao longo desta dissertação foi todo ele desenvolvido e testado em ambiente laboratorial.

Tendo em vista a validação do sistema proposto, foram utilizados dois dispositivos móveis, cujas características técnicas se encontram nas tabelas 6.2 e 6.3. Além disso, recorreu-se à criação de três documentos com as informações dos três estádios a mapear. O efeito visual chamado de “onda” encontra-se implementado e o sistema possui capacidade para o realizar, em qualquer um dos locais definidos.

A aplicação móvel encontra-se implementada para o sistema operativo Android. Construiu-se a aplicação móvel cuja interface se apresenta na figura 6.7. Em particu-

6.7. Resultados obtidos

lar, a figura 6.7 representa a interação de um utilizador que pretende ser localizado de forma explícita. Pelo facto de serem utilizados dois dispositivos móveis, com características diferentes, verifica-se que aplicação se adapta aos diferentes ecrãs.

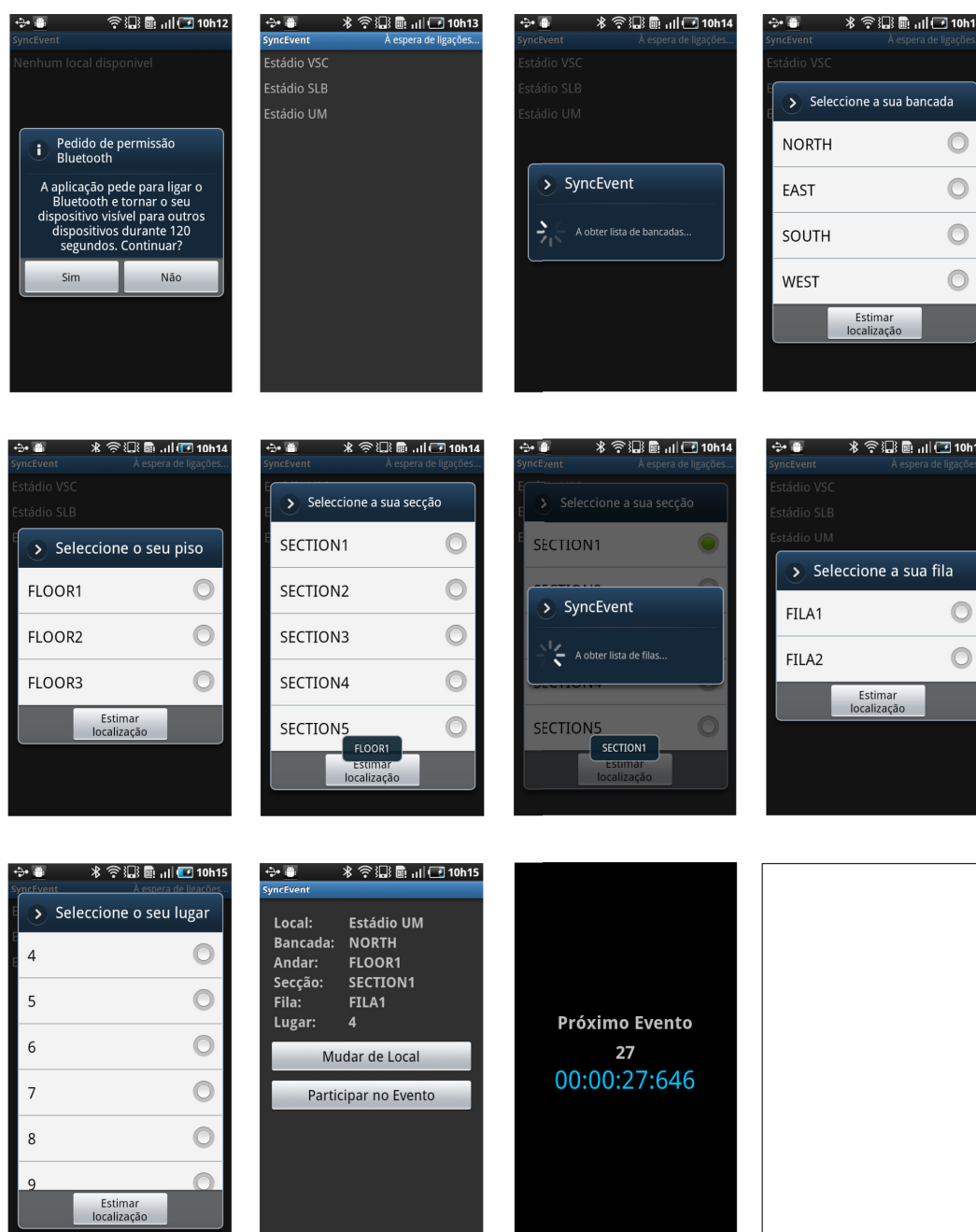


Figura 6.7: Aplicação móvel do SyncEvent.

Os dispositivos móveis, mesmo os mais recentes, possuem limitações energéticas. Desta forma, um dos aspetos que teve direito a alguma ponderação foi a transferência de informação entre os dispositivos móveis e o servidor. Tendo em vista a redução do impacto do elevado número de mensagens transferidas entre os intervenientes do sistema, as aplicações utilizam o *ProtoBuf*. Este mecanismo compacta os dados transmitidos de forma a que ocupem menos espaço sem qualquer dano na informação.

A figura 6.8, ilustra a aplicação do servidor. O *layout* do servidor possui uma interface gráfica, onde é possível ligar/desligar, e uma listagem dos acontecimentos mais importantes do sistema. O servidor foi implementado no sistema operativo Mac OS X (v10.6.8), utilizando o Eclipse como Integrated Development Environment (IDE) para o desenvolvimento. No entanto, uma vez que o servidor foi desenvolvido em Java, facilmente é executado em qualquer outro sistema operativo.

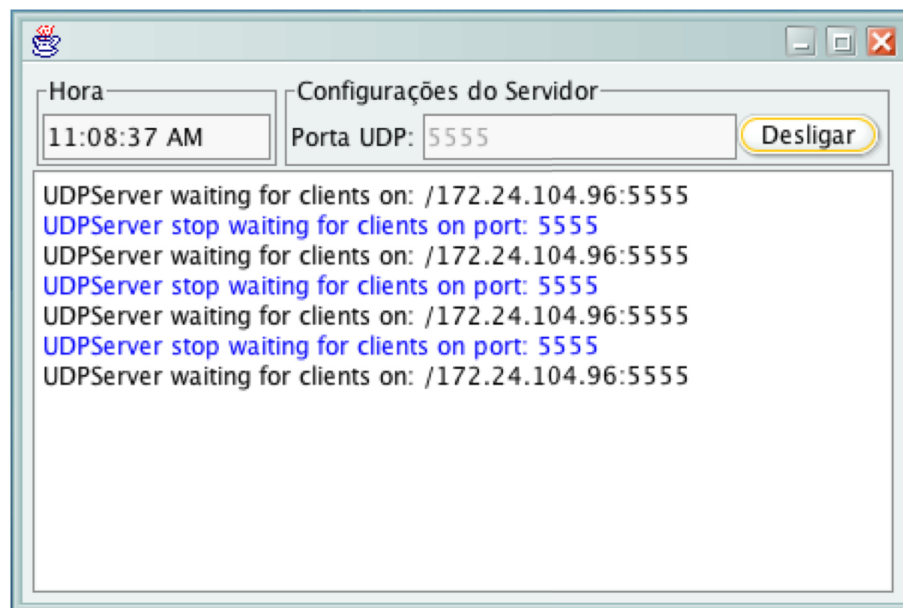


Figura 6.8: Servidor do SyncEvent.

Capítulo 7

Conclusões e trabalho futuro

Este capítulo conclui a presente dissertação. Ao longo deste capítulo, é realizada uma reflexão sobre os objetivos e implicações dos resultados obtidos, apresentam-se as limitações do estudo e algumas recomendações para uma possível investigação futura.

7.1 Conclusões do projeto

Neste documento foi apresentado um sistema destinado a produzir um determinado efeito coordenado, entre os telefones móveis dos espectadores presentes num evento. Em relação aos objetivos propostos:

- Foi construído um sistema com capacidade para propagar um evento pelos utilizadores, que possuam a aplicação SyncEvent instalada nos seus dispositivos móveis, e que desejem participar no evento;
- O primeiro evento disponibilizado pelo SyncEvent é análogo ao efeito de “onda”, realizado habitualmente pelos espectadores de um entretenimento público.
- O evento disponibilizado ocorre de forma sincronizada, sendo que este processo se realiza no instante em que os utilizadores tomam a iniciativa de participar num evento.
- Em relação à localização dos dispositivos, coloca-se à disposição dos utilizadores a possibilidade de serem localizados de forma explícita ou estimada.

Uma das principais ilações que emerge do estudo do problema é que, apesar do Bluetooth já se encontrar bastante desenvolvido, o estabelecimento de ligações com outros dispositivos requer demasiado tempo. Além disso, para a utilização do Bluetooth em projetos deste género, devem ser considerados fatores como: interferências causadas pela presença de um grande número de dispositivos na mesma área de cobertura; complexidade da formação de grandes redes; partição da rede e diversidade de dispositivos. No entanto, tal como os dispositivos móveis, as tecnologias que estes integram encontram-se em constante evolução. O Bluetooth encontra-se em desenvolvimento progressivo sendo possível que, num futuro próximo, o estabelecimento de ligações venha a consumir menos tempo do que na atualidade.

Durante a realização deste trabalho foram ultrapassados problemas e desafios, abrindo-se uma nova possibilidade de interação entre os espectadores de um determinado evento. Mais do que isso, foi possível obter experiências e conhecimentos de um vasto conjunto de tecnologias. Foi ainda obtida a noção das dificuldades e do estado atual do desenvolvimento de aplicações direcionadas a dispositivos móveis.

7.2 Limitações da aplicação desenvolvida

Apesar de todo o empenho empregue na realização do trabalho, reconhece-se a existência de pontos que não se encontram ponderados pelo estudo. Existe por isso, um conjunto de restrições e condições na utilização do sistema.

Antes de mais, a aplicação móvel foi desenvolvida apenas para o sistema operativo Android. Apesar da elevada quota de mercado que este possui, não é possível desta forma alcançar a generalidade de dispositivos.

Em relação ao processo de localização estimada, o servidor atribui um lugar vazio ao utilizador. No entanto, não tem capacidade para alterar o utilizador de lugar, caso esse lugar seja ocupado, de forma explícita, por terceiros.

Devido à limitação de tempo, não foi efetuada uma interface que permita a introdução de novos eventos, quando a aplicação já se encontra em funcionamento. Assim, é necessário parar a aplicação, definir um novo evento e reativar o servidor.

O uso do sistema prende-se somente pelo preceito de identificação da localização dos dispositivos. Assim, o sistema não funciona em espaços livres, onde não se identifica os lugares ocupados pelos espectadores. Todavia, devido à falta de testes num ambiente mais autêntico, não é possível conhecer as verdadeiras limitações do sistema.

7.3 Evolução no mecanismo de posicionamento

Numa fase final, surgiu um melhoramento a efetuar à arquitetura do sistema desenvolvido. A aquisição da localização dos utilizadores, realizada de forma estimada, requer a comunicação entre um ou mais dispositivos através do Bluetooth. Para além da necessidade de serem estabelecidas ligações entre os dispositivos através do Bluetooth, o que consome demasiado tempo, existe a necessidade de se obterem permissões para o acesso à informação, o que pode ser um incómodo para alguns utilizadores.

A tecnologia Bluetooth permite a deteção de outros dispositivos que se encontrem ao seu alcance. Um utilizador que pretenda ser localizado de forma estimada pode detetar outras interfaces Bluetooth e, em caso de sucesso, informar o servidor dos dispositivos que se encontram próximos. Com a informação da vizinhança, quando um utilizador optar por ser localizado de forma estimada, o servidor pode atribuir um lugar que se encontre livre, perto daqueles que consegue atingir através do Bluetooth.

Com esta evolução no mecanismo de posicionamento de forma implícita, elimina-se a necessidade do estabelecimento de ligações Bluetooth, resultando num menor tempo despendido na localização de cada dispositivo. Esta evolução torna-se mais transparente para os utilizadores permitindo salvaguardar os utilizadores com menor paciência, que não pretendem ser incomodados com constantes pedidos de ligação.

Evidentemente que este melhoramento ao processo de localização estimada, incorre numa maior sobrecarga computacional no servidor, devido à necessidade da implementação de mecanismos de localização mais elaborados. Contudo, o sistema atual pode adaptar-se facilmente a esta evolução, quer no que diz respeito à utilização do endereço MAC das interfaces Bluetooth, quer na capacidade de atribuir um lugar livre na proximidade dos dispositivos que consegue alcançar.

7.4 Trabalho futuro

Este trabalho deixa ainda muito por explorar em relação à criação de eventos sincronizados em sistemas móveis fisicamente próximos. Desde logo, fica clara a necessidade da realização de testes efetivos, que permitam verificar os verdadeiros limites do sistema desenvolvido. Através de uma análise dos resultados obtidos, deve ser verificada a necessidade da introdução de mecanismos de escalabilidade, de forma a aumentar a eficiência do sistema perante aumentos significativos no número de utilizadores.

Propõe-se uma otimização do código desenvolvido de modo a facilitar a inclusão de novos eventos. Em paralelo, seria interessante disponibilizar diversos eventos e permitir que seja o utilizador a seleccionar os eventos em que pretende participar. Seria de igual modo interessante que os eventos pudessem ser iniciados, ou até mesmo criados pelos utilizadores do sistema.

De modo a maximizar a disponibilidade do sistema, propõe-se o desenvolvimento da aplicação SyncEvent para diferentes sistemas operativos (Symbian, iOS, e WP7). Nesta vertente, prevê-se também que a existência de recompensas na utilização do sistema, poderá impulsionar o interesse dos utilizadores em participar.

Durante a realização deste trabalho, foram ignoradas questões de segurança e privacidade. No entanto, estas questões devem ser consideradas, não só para o êxito da aplicação, mas também para o seu bom funcionamento. Sendo que, para a evolução do sistema, até quem sabe para uma implementação comercial, considera-se imprescindível a introdução destes pontos em estudos futuros.

Finalmente seria interessante analisar detalhadamente a usabilidade do sistema. Sendo que se sugere a realização de uma série de questões a um grupo de pessoas, a fim de se obter noção da opinião pública sobre o sistema. Seria importante saber se as pessoas ponderam a utilização da aplicação e, admitindo alguma curiosidade, que tipo de eventos gostavam de ver disponíveis e onde desejavam usufruir do sistema. Para trabalho futuro, seria ainda de elevada importância verificar se a interface gráfica da aplicação móvel se adapta convenientemente ao propósito para o qual foi construída.

Referências e bibliografia

- Adobe Inc. (2011). Adobe AIR. Consultado em 4 de abril de 2011, em <http://www.adobe.com/products/air/>.
- Aiello, M., de Jong, R., & de Nes, J. (2009). Bluetooth broadcasting: How far can we go? an experimental study. In *Pervasive Computing (JCPC), 2009 Joint Conferences on*, (pp. 471–476).
- Android (2011). Android Open Source Project. Consultado em 27 de setembro de 2011, em <http://developer.android.com/index.html>.
- Asthana, S., & Kalofonos, D. (2005). The problem of bluetooth pollution and accelerating connectivity in bluetooth ad-hoc networks. In *Pervasive Computing and Communications, 2005. PerCom 2005. Third IEEE International Conference on*, (pp. 200 –207).
- Barrera, D., & Oorschot, P. (2011). Secure software installation on smartphones. *Security & Privacy, IEEE, PP(99)*, 1.
- Blom, S., Book, M., Gruhn, V., Hrushchak, R., & Kohler, A. (2008). Write once, run anywhere a survey of mobile runtime environments. In *Grid and Pervasive Computing Workshops, 2008. GPC Workshops '08. The 3rd International Conference on*, (pp. 132 –137).
- Bluetooth SIG (2009). Specification of the Bluetooth System. *Version 3.0 + HS*.
- Bluetooth SIG (2011). The official bluetooth web site. Consultado em 10 de setembro de 2011, em <http://www.bluetooth.com/Pages/Bluetooth-Home.aspx>.
- Chlamtac, I., Conti, M., & Liu, J. J. N. (2003). Mobile ad hoc networking: imperatives and challenges. *Ad Hoc Networks*, 1(1), 13–64.

- De Vriendt, J., Laine, P., Lerouge, C., & Xu, X. (2002). Mobile network evolution: a revolution on the move. *Communications Magazine, IEEE*, 40(4), 104–111.
- Farkas, I., Helbing, D., & Vicsek, T. (2002). Mexican waves in an excitable medium. *Nature*, 419(6903), 131–132.
- Ferro, E., & Potorti, F. (2005). Bluetooth and Wi-Fi wireless protocols: a survey and a comparison. *IEEE Wireless Communications*, 12(1), 12–26.
- Frodigh, M., Johansson, P., & Larsson, P. (2000). Wireless ad hoc networking: the art of networking without a network. *Ericsson Review*, No.4, 248–263.
- Gartner (2011). Technology research & business leader insight. Consultado em 30 de março de 2011, em <http://www.gartner.com/technology/home.jsp>.
- Gavalas, D., & Economou, D. (2011). Development platforms for mobile applications: Status and trends. *Software, IEEE*, 28(1), 77–86.
- Ghosh, A., Ratasuk, R., Mondal, B., Mangalvedhe, N., & Thomas, T. (2010). Lte-advanced: next-generation wireless broadband technology [invited paper]. *Wireless Communications, IEEE*, 17(3), 10–22.
- Gligoric, N., Dejanovic, I., & Krco, S. (2011). Performance evaluation of compact binary XML representation for constrained devices. In *Distributed Computing in Sensor Systems and Workshops (DCOSS), 2011 International Conference on*, (pp. 1–5). IEEE.
- Guo, C., Wang, H. J., & Zhu, W. (2007). Smart-Phone attacks and defenses. Consultado em 20 de agosto de 2011, em <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.134.3703>.
- Haas, Z., Halpern, J., & Li, L. (2002). Gossip-based ad hoc routing. In *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3, (pp. 1707–1716 vol.3).
- Hammershoj, A., Sapuppo, A., & Tadayoni, R. (2010). Challenges for mobile application development. In *Intelligence in Next Generation Networks (ICIN), 2010 14th International Conference on*, (pp. 1–8).

- Handurukande, S., Ganguly, S., & Bhatnagar, S. (2006). Fast bluetooth service discovery for mobile peer-to-peer applications. <http://www.sigmobility.org/mobisys/2006/posters/Handurukande.pdf>.
- Ho, J. (2006). *Bluetooth broadcasting*. Master of science in computer science, Norwegian University of Science and Technology. <http://www.idi.ntnu.no/grupper/su/su-diploma-2006/ho-2006.pdf>.
- Hollister, S. (2011). Adobe AIR developer demonstration: one game, five platforms, all the same code. Consultado em 19 de abril de 2011, em <http://engadget.com/2010/04/05/adobe-air-developer-demonstration-one-game-five-platforms-all/>.
- IEEE 802.11 (2007). IEEE Standard - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *IEEE Std 802.11-2007 (Revision of IEEE Std 802.11-1999)*, (pp. C1 –1184).
- Khelil, A. (2007). *A generalized broadcasting technique for mobile ad hoc networks*. Ph.D. thesis, Universität Stuttgart, Holzgartenstr. 16, 70174 Stuttgart. <http://elib.uni-stuttgart.de/opus/volltexte/2007/3120>.
- Khelil, A., Marrón, P. J., Becker, C., & Rothermel, K. (2005). Hypergossiping: A generalized broadcast strategy for mobile ad hoc networks. In *In Proceedings of The 2005 Conference on Communication in Distributed Systems (KiVS)*, (pp. 531–546).
- Kumar, K., & Lu, Y. (2011). Cloud computing for mobile users. *Computer*, PP(99).
- Lasassmeh, S., & Conrad, J. (2010). Time synchronization in wireless sensor networks. In *IEEE SoutheastCon 2010, Proceedings of the*, (pp. 242 –245).
- Li, X., Gani, A., Salleh, R., & Zakaria, O. (2009). The future of mobile wireless communication networks. In *Communication Software and Networks, 2009. ICCSN '09. International Conference on*, (pp. 554 –557).
- Livingston, A. (2004). Smartphones and other mobile devices: The swiss army knives of the 21st century. *EDUCAUSE Quarterly*, 27(2), 46–52.
- Oliver, E. (2009). A survey of platforms for mobile networks research. *SIGMOBILE Mob. Comput. Commun. Rev.*, 12, 56–63.

- Oracle Corporation (2011). Java ME - the most ubiquitous application platform for mobile devices. Consultado em 1 de abril de 2011, em <http://www.oracle.com/technetwork/java/javame/>.
- Paus, A. (2007). *Near field communication in cellphones*. Ph.D. dissertation, Ruhr-Universitat Bochum.
- Pinto, P. (2011). *Pplware*. Consultado em 20 de agosto de 2011, em <http://www.pplware.sapo.pt/networking/tecnologias-1g-2g-2-5-g-3g-e-4g-sabe-a-diferenca/>.
- Rao, L. (2011). Android OS sales trump iOS and RIM, grew 888 percent in 2010. Consultado em 30 de março de 2011, em <http://techcrunch.com/>.
- RFID Journal (2011). RFID FAQs. Consultado em 15 de agosto de 2011, em <http://www.rfidjournal.com/faq>.
- Romero, J. J. (2011). Smartphones: The pocketable PC - IEEE spectrum. Consultado em 4 de maio de 2011, em <http://spectrum.ieee.org/telecom/wireless/smartphones-the-pocketable-pc>.
- Samsung (2011). Samsung. Consultado em 3 de junho de 2011, em <http://galaxys2.samsungmobile.com/html/specification.html>.
- Schiller, J. (2003). *Mobile Communications*. Addison-Wesley, second edition ed.
- Scholz, M., Bregulla, F., & Hinze, A. (2007). Using physical clocks for replication in MANETs. In *Pervasive Computing and Communications Workshops, 2007. PerCom '07. Fifth Annual IEEE International Conference on*, (pp. 114–119).
- Spinellis, D. (2006). Choosing a programming language. *IEEE Software*, 23, 62–63.
- Thamrin, T., & Sahib, S. (2009). The inquiry and page procedure in bluetooth connection. In *International Conference of Soft Computing and Pattern Recognition, 2009. SOCPAR '09*, (pp. 218–222). IEEE.
- Vitsas, V., & Barker, P. (2003). IrDA infrared wireless communications: protocol throughput optimization. *Wireless Communications, IEEE*, 10(2), 22–29.
- Want, R. (2011). Near field communication. *IEEE Pervasive Computing*, 10(3), 4–7.

- Williams, B., & Camp, T. (2002). Comparison of broadcasting techniques for mobile ad hoc networks. In *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, (pp. 194–205).
- Wong, K. Y. (2010). Cell phones as mobile computing devices. *IT Professional*, 12(3), 40–45.
- Xiao, J., Liu, Z., Yang, Y., Liu, D., & Han, X. (2011). Comparison and analysis of indoor wireless positioning techniques. In *Computer Science and Service System (CSSS), 2011 International Conference on*, (pp. 293 –296).
- Ziegler, C. (2010). LG optimus 2X: first dual-core smartphone launches with android. Consultado em 10 de março de 2011, em <http://www.engadget.com/2010/12/15/lg-optimus-2x-first-dual-core-smartphone-launches-with-android/>.
- Ziegler, C. (2011). IDC says 100.9M smartphones sold in fourth quarter, PCs outsold for first time. Consultado em 10 de março de 2011, em <http://engadget.com/2011/02/08/idc-says-100-9m-smartphones-sold-in-fourth-quarter-pcs-outsold/>.
- ZigBee Alliance (2011). ZigBee. Consultado em 15 de agosto de 2011, em <http://www.zigbee.org/>.

Apêndices

Apêndice A

Protocolo de Comunicação

Na listagem A.1 apresenta-se o conteúdo do documento utilizado para descrever as estruturas de dados utilizadas para a comunicação entre os dispositivos móveis e o servidor do SyncEvent.

```
1 package com.SyncEvent.Protocol;
2
3 option java_package = "com.SyncEvent.Protocol";
4 option java_outer_classname = "SyncEventProto";
5
6 option optimize_for = SPEED;
7
8 message LocationMessage {
9     optional string UUID = 1;
10    optional string Local = 2;
11    optional string Stand = 3;
12    optional string Floor = 4;
13    optional string Section = 5;
14    optional string Row = 6;
15    optional string Seat = 7;
16 }
17
18 message EventMessage {
19     optional string EventId = 1;
20     optional string EventDesc = 2;
21     optional string DeviceId = 3;
22     optional string DeviceTime = 4;
23     optional string SignalTime = 5;
```

```

24     repeated string EventTime = 6;
25 }
26
27 message Message {
28     enum MessageType {
29         LOCALS_REQUEST = 0;
30         LOCALS_REPLY = 1;
31         STANDS_REQUEST = 2;
32         STANDS_REPLY = 3;
33         FLOORS_REQUEST = 4;
34         FLOORS_REPLY = 5;
35         SECTIONS_REQUEST = 6;
36         SECTIONSS_REPLY = 7;
37         ROWS_REQUEST = 8;
38         ROWS_REPLY = 9;
39         SEATS_REQUEST = 10;
40         SEATS_REPLY = 11;
41         IMP_SELECT = 12;
42         IMP_REPLY = 13;
43         SELECT_REPLY = 14;
44         SEAT_SELECT = 15;
45         BT_SEAT_REQ = 16;
46         BT_SEAT_REP = 17;
47         EVENT_JOIN = 18;
48         EVENT_REPLY = 19;
49     }
50 }
51
52     optional MessageType type = 1;
53     optional LocationMessage locationMessage = 2;
54     optional EventMessage eventMessage = 3;
55 }

```

Listagem A.1: Estruturas de dados definidas para a comunicação entre os intervenientes do SyncEvent.

Apêndice B

Contributo

Marques, N. & Meneses, F. (2011). Synchronized Events in Mobile Systems Physically Nearby. In *Indoor Positioning and Indoor Navigation (IPIN), 2011. International Conference on*.

URL http://ipin2011.dsi.uminho.pt/PDFs/Poster/130_Poster.pdf

Synchronized Events in Mobile Systems Physically Nearby

Nelson Marques
a48064@alunos.uminho.pt

Filipe Meneses
meneses@dsi.uminho.pt

Motivation

Normally in a football stadium, spectators raise their arms in a coordinated movement to produce a visual effect called “wave”.

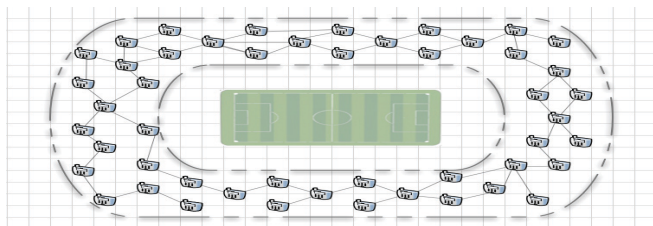
The purpose of this study is to develop a system that allows spectators at an event to be able to accomplish a particular coordinated effect through their mobile phones.

The difference to the model with the arms in the air is that now the wave is created, for example, by turns on and off in a coordinated manner, the display or the keyboard of the mobile devices.

IMAGINE THE POSSIBILITY OF DOING A MEXICAN WAVE WITH YOUR MOBILE PHONE.

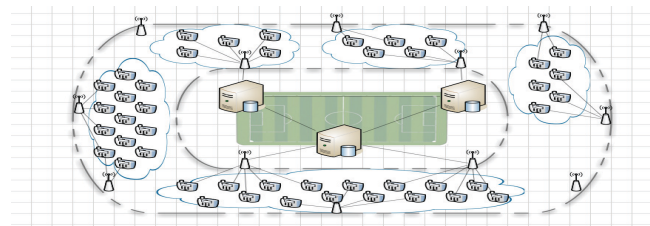
Ad hoc Approach (With Bluetooth)

- One of the disadvantages of combining the BT technology with this model is the great complexity of the nodes forming the network.
- Despite the maturity of BT, the initial connection, even in the newer versions, are still consuming too much and even after a connection is establish, each device has it's way of dealing with incoming messages.
- Besides that, the union of the *ad hoc* model with BT technology is likely to partitioning the network.



Approach With Infrastructure

- The approach with infrastructure reduces the complexity in the applications of the devices.
- Although it involves the creation of an infrastructure, today, their costs are reduced.
- A major advantage of infrastructure based solution is the existence of a central system that avoids the activation of multiple events simultaneously and the creation of multiple networks (which may easily happen in *ad hoc* approach).



System Overview

- To participate in a particular event, users can communicate with the server via a Wi-Fi infrastructure that will be found in the area or directly through other Internet access technology that mobile devices have such as UMTS or HSPA.
- The server holds a list of users and sends a message to each of them with the event information. That message includes the type of event (single wave, double wave, etc.) and the execution time.
- The execution time is computed according to the user's location in the stadium. That time is different for each place, depending on user's location.

